



НИИ МИВАР

Под редакцией
О.О. Варламова

**ПРАКТИКУМ
ПО СОЗДАНИЮ МИВАРНЫХ
ЭКСПЕРТНЫХ СИСТЕМ**

Москва
2016

НИИ МИВАР

**ПРАКТИКУМ ПО СОЗДАНИЮ
МИВАРНЫХ ЭКСПЕРТНЫХ
СИСТЕМ**

Учебное пособие под редакцией
Варламова О.О.

Москва
2016

УДК 004.82+ 007.52
ББК 32.813
В18

Авторы: Варламов О.О., Чибирова М.О., Хадиев А.М., Антонов П.Д., Сергушин Г.С., Шошев И.А., Назаров К.В.
Практикум по созданию миварных экспертных систем. Учебное пособие / под ред. О.О. Варламова. — М.: Изд-во НИИ МИВАР, 2016. — 184 с.

Рассмотрены методологические и прикладные вопросы создания логического искусственного интеллекта, систем управления базами знаний и логического вывода. Рассматривается созданный на основе миварных технологий программный комплекс «Конструктор экспертных систем миварный (КЭСМИ) Wi!Mi РАЗУМАТОР» (версии 2.1), который представляет собой удобный инструмент для разработки интеллектуальных информационных систем. Приведены подробное описание работы КЭСМИ, примеры создания экспертных систем и несколько лабораторных работ. Читатель, изучив «Практикум по созданию миварных экспертных систем», сможет самостоятельно создавать экспертные системы с помощью КЭСМИ.

Учебное пособие предназначено для студентов, бакалавров, магистрантов, аспирантов, изучающих методы искусственного интеллекта, а также для пользователей, экспертов и специалистов, создающих миварные модели, экспертные системы, автоматизированные системы управления и системы поддержки принятия решений.

Рецензенты:

Докт. физ.-мат. наук, проф. М.М. Никитин;
Докт. техн. наук., проф. С.М. Чудинов.

Учебное пособие
© Варламов О.О., 2016
© ООО МИВАР, 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ	
1.1. Искусственный интеллект. Определение искусственного интеллекта и его приложения.....	8
1.2. Роль представления и обработки информации в искусственном интеллекте.....	10
1.3. Познающе-диагностические автоматизированные информационные системы и сложные предметные области.....	13
1.4. Технологии и достижения ИИ.....	16
1.5. Представление знаний в ИИ.....	30
1.6. Анализ существующих парадигм и моделей обработки данных.....	34
2. МИВАРНЫЕ ТЕХНОЛОГИИ	
2.1. Основы миварного подхода.....	40
2.2. Почему миварная технология обработки информации быстрее, чем продукции.....	54
2.3. Для каких систем создан миварный подход.....	55
2.4. Сравнительный анализ миварного подхода и технологий ИИ.....	60
3. ВОЗМОЖНОСТИ И ПРЕИМУЩЕСТВА МИВАРНОГО ПОДХОДА	
3.1. Возможности миварного метода.....	62
3.2. Преимущества миварного подхода перед семантическими сетями и продукциями.....	63
4. МИВАРНЫЙ МЕТОД ЛОГИКО- ВЫЧИСЛИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ	
4.1. Три этапа миварной обработки информации.....	69
4.2. Теоретический расчет вычислительной сложности миварного метода обработки.....	81
4.3. Научно-популярная иллюстрация применения миварного подхода.....	82

4.4. Сказка: паровоз, искусственный интеллект и Старик Хоттабыч.....	86
5. РУКОВОДСТВО ПО КЭСМИ РАЗУМАТОР WIMI 2.1	
5.1. Описание программы.....	89
5.2. Область применения.....	89
5.3. Назначения и условия применения.....	90
5.4. Подготовка к работе.....	90
5.5. Работа с программой.....	92
5.6. Возможные ошибки.....	114
5.7. Рекомендации по освоению.....	115
6. ЛАБОРАТОРНЫЕ РАБОТЫ	
6.1. Описание процесса выполнения лабораторных работ.....	116
6.2. Лабораторная работа № 1.....	119
6.3. Лабораторная работа № 2.....	127
6.4. Лабораторная работа № 3.....	134
6.5. Лабораторная работа № 4.....	144
6.6. Лабораторная работа № 5.....	151
7. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ.....	163
СПИСОК СОКРАЩЕНИЙ.....	165
СПИСОК ЛИТЕРАТУРЫ.....	166

ВВЕДЕНИЕ

Миварный подход относится к научному направлению «искусственный интеллект» и развивается уже более 30 лет [1–4]. В исследованиях применяются два основных сокращения, используемые на английском и русском языках: **MIVAR** (Multidimensional Informational Variable Adaptive Reality) или **МИВАР** (Многомерная Информационная Варьирующая Адаптивная Реальность). Вместе с тем миварный подход объединяет и другие научные области, среди которых информатика и дискретная математика, включая теорию графов, сети Петри, теорию предикатов. Рассматриваемый подход, включающий в себя технологию представления знаний, а также технологию их обработки, может использоваться при разработке экспертных систем, систем логического вывода, в робототехнике, много-агентных системах, в системах распознавания образов, системах понимания естественного языка и т.д.

В XX в. стали активно создаваться различные экспертные системы [6, 10, 14]. Такие системы, концентрируя знания специалистов, дают возможность тиражировать их для консультаций пользователей. Однако в силу сложности формализованного описания предметных областей и того, что системы логического вывода не могли обрабатывать большое количество объектов или правил, такие экспертные системы разрабатывались для отдельных узконаправленных предметных областей. Параллельно с этим развивались интеллектуальные пакеты прикладных программ, которые позволяли решать задачи из различных областей, где требовались конструирование алгоритмов решения задач и вычисления в автоматизированном режиме. Обе разработки предопределили возникновение экспертных систем нового поколения, появление которых дает возможность автоматизировать решение различных сложных интеллектуальных задач и повышает конкурентоспособность их пользователей. Одной из современных разработок в данной

сфере можно назвать созданный на основе миварных технологий программный комплекс КЭСМИ Wi!Mi РАЗУМАТОР, который позволяет создавать экспертные системы в различных предметных областях, в том числе содержащих более 5 млн объектов и правил, а также находить алгоритмы решения задач в этих предметных областях за доли секунды на обычных персональных компьютерах и ноутбуках.

Сама идея создания миварных технологий появилась в 1985 г. в процессе решения задачи о применении сетей Петри для автоматического решения геометрических задач о треугольниках, где по разным параметрам (углам, сторонам треугольника, высотам и т.д.) с использованием существующих формул всегда можно найти другие параметры, например, площадь, периметр и др. Такая разработка должна была применяться как основа для создания прототипов интеллектуальных пакетов прикладных программ различного назначения. Однако в процессе изучения сетей Петри были выявлены ограничения и неполная пригодность данного формализма даже для решения таких простых задач, как расчет параметров треугольника, в связи с чем потребовалось создание нового формализма, позже получившего название миварного подхода.

Здесь стоит особенно отметить профессора Ю.Г. Ростовцева [15], который поддержал идею миварного подхода, подчеркнув что, рассматривая любые теоретические вопросы, нельзя забывать о важности решения именно практических задач и о необходимости разработки новых инструментов. При этом всегда надо помнить, что любой математический формализм — это ограничения на предметную область, а значит упрощение реальной жизни, и следовательно, надо вовремя переходить к новым инструментам. Миварные технологии можно отнести к одному из таких новых инструментов. Эти технологии, как подтверждает опыт преподавания дискретной математики студентам ряда московских вузов, по своей сути являются простыми для освоения и понимания.

В учебном пособии изложены теоретические основы миварного подхода, которые за прошедшее время получили свое подтверждение и практическое развитие в различных предметных областях. Разнообразие материала в этой книге объясняется универсальностью миварного подхода и возможностью его применения ко многим областям искусственного интеллекта.

Отдельно стоит подчеркнуть, что необходимо помнить и о том, что кроме положительной стороны стремительного развития информатики и искусственного интеллекта (ИИ) есть и обратная сторона медали. К отрицательным сторонам можно отнести многие проблемы информационной безопасности и технической защиты информации. Есть и другие потенциальные отрицательные последствия, о которых достаточно широко известно по катастрофическим прогнозам фантастов. Как бы скептически человечество ни относилось к такого рода перспективам, но учитывать и изучать подобные вероятности необходимо заранее. При создании систем ИИ обязательно должен проводиться анализ отрицательных последствий и разработка предложений по их недопущению.

В литературе [1–4, 22–109] показано, что на основе миварного подхода создан логический искусственный интеллект (ЛИИ), который позволит перейти на качественно новый уровень научных достижений по всем направлениям ИИ: представление знаний; планирование целесообразного поведения; общение с ЭВМ на естественном языке; распознавание зрительных образов и понимание речевой информации. В учебном пособии описывается созданный на основе миварных технологий программный комплекс КЭСМИ «Wi!Mi РАЗУМАТОР», который дает возможность практически любому человеку самостоятельно создать свою собственную экспертную систему и материализовать свои знания.

1. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

1.1. Искусственный интеллект. Определение искусственного интеллекта и его приложения.

Большой интерес у исследователей вызывает вопрос: **создан ли уже искусственный интеллект (ИИ)?**

Итоги дискуссий и разнообразных круглых столов на научных конференциях однозначно показывают, что ответы на него прямо противоположны: от категоричного отрицания самой возможности создания ИИ, до утверждений, что он уже создан. Такая ситуация вызвана отсутствием устоявшегося определения термина «искусственный интеллект», что обуславливает необходимость введения шкалы измерения интеллектуальности различных систем. Постепенно именно такой подход получает все большую поддержку научного сообщества, поскольку он позволяет свести споры о создании ИИ к единой базе и точке отсчета.

По мере развития ИИ компьютерам передавалось все большее количество функций и возможностей (под словом «компьютер» подразумеваются и программы, и аппаратные устройства, которые создаются человеком). При этом наблюдается закономерность: как только программистам удавалось создать новую автоматическую систему решения творческих человеческих задач, некоторые ученые переводили эту задачу в разряд нетворческих и продолжали утверждать, что ИИ не существует. За последние 50 лет развития компьютеров осталось не так уж много задач, которые люди все еще решают лучше. Таким образом, компромисс при определении термина «искусственный интеллект» возможен при следующих условиях:

– широкое признание того, что область ИИ и сам ИИ — это усилитель человеческих мыслительных возможностей, и он

нужен прежде всего для автоматизации различных функций при работе с информацией;

- введение некой шкалы автоматизации, активности или интеллектуальности человеко-машинных систем и компьютеров;
- признание того, что некие достаточно простые формы усиления мышления или его автоматизации уже созданы, т.е. некие прототипы ИИ уже существуют. Доказательством этого тезиса служит то, что современный человек уже не мыслит своей жизни без компьютеров, которые в любом случае создают преимущества для своих владельцев.

Стоит напомнить, что изначально область ИИ занималась автоматизацией разумной деятельности человека. Все соответствующие западные и российские исследования направлены именно на это (миварные технологии, описанные в данной книге, преследуют ту же цель). Поэтому несмотря на различные споры о том, что же является искусственным интеллектом, в данном контексте целесообразно будет привести следующее формальное определение ИИ.

Искусственный интеллект — область информатики, занимающаяся автоматизацией разумной деятельности человека.

Под термином *системы искусственного интеллекта* в данной книге понимаются активные логически рассуждающие системы (ЛРС).

Стоит отметить, что представленное определение не раскрывает проблему создания полностью автоматического ИИ, который по своим возможностям будет превосходить как «типового» человека, так и самого «умного» (данную трактовку термина применяют противники возможности создания полноценного ИИ).

1.2. Роль представления и обработки информации в искусственном интеллекте

Информатизация — это применение и развитие компьютерных технологий. В современном состоянии это, в частности, означает повсеместное использование территориально распределенных гетерогенных компьютерных сетей и систем. Необходимо сразу же выделить два принципиально разных уровня информатизации:

- физический (оборудование, каналы связи и т.п.);
- логический (программное обеспечение, алгоритмы, данные и т.п.).

На логическом уровне, по существу, компьютеры материализуют человеческие мысли, т.е. выполняют написанные определенные алгоритмы и программы. Алгоритмы могут быть самыми разными. Человек, разрабатывая алгоритм и создавая программу, «отделяет» от себя свои мысли, дает им самостоятельную жизнь, но потом может воспользоваться результатами работы созданных программ. До тех пор пока не создан полноценный автономный искусственный интеллект, компьютер является усилителем мыслительной деятельности человека. Принципиальное свойство информатизации — усиливать человеческие интеллектуальные способности. Таким образом, можно прийти к выводу о том, что ни одна система без человека не работает, т.е. если мыслей и интеллекта нет, то и усиливать нечего.

Основные процессы, которые автоматизирует информатика — это сбор, передача, обработка, хранение и представление данных (информации, сведений и т.п.). По всем этим пяти направлениям компьютеры усиливают и/или расширяют человеческие возможности (способности). Кроме того, с помощью компьютера человек может создавать новую информацию (данные). Стоит отметить, что принято различать понятия «информация», «сведения», «данные» и близкие к ним, однако в настоящей книге в этом нет необходимости, поэтому все названные термины будут исполь-

зоваться как синонимы. Далее рассмотрим подробнее упомянутые выше процессы.

1. Сбор информации. Компьютерные системы и сети позволяют человеку собирать (получать) практически всю документированную и переведенную в электронный вид информацию, т.е. возможности человека по сбору информации становятся почти безграничными.

2. Передача информации. Современные средства компьютерной телекоммуникации дают возможность передавать информацию в очень больших объемах практически в реальном времени на любые расстояния и многим получателям. Особо надо подчеркнуть, что передача позволяет взаимодействовать большому количеству людей почти одновременно. Есть ограничения по скорости передачи данных, но в целом даже видеоизображение уже может передаваться в хорошем качестве, быстро и без особых финансовых затрат.

3. Обработка информации. В этом направлении все зависит от формализуемости и наличия четких алгоритмов обработки информации и их вычислительной сложности. Все операции, для которых разработаны алгоритмы, могут быть выполнены на компьютерах, но за разное время. Можно условно разделить все задачи на три класса: 1) не придуманы алгоритмы решения; 2) алгоритмы известны, но время их выполнения выходит за разумные рамки и может составлять тысячелетия (например, NP-полные задачи); 3) алгоритмы решения известны и выполняются достаточно быстро. Первые два класса задач в настоящее время не могут быть решены в полном объеме. Таким образом, возможности информатики по обработке данных, конечно, ограничены, но все же компьютеры позволяют существенно усиливать человеческие способности и проводить обработку огромных массивов информации достаточно быстро. Пожалуй, именно в обработке остались большие резервы для усиления человеческого мышления (по возможности и по мере необходимости).

4. Хранение информации. При соблюдении определенных условий хранить можно практически бесконечные объемы информации. Отдельный вопрос — это организация оперативно-го доступа к требуемой информации, но здесь еще очень много нерешенных научно-технических проблем, что связано с обработкой информации. Отдельная проблема — это создание эволюционных многомерных баз данных для непрерывного накопления информации с изменяемой структурой хранения. Для решения таких проблем можно применять, например, миварное информационное пространство, позволяющее в едином формате хранить и данные, и правила их обработки (процедуры, программы и т.п.).

5. Представление информации. В этом направлении все зависит от требуемых форм представления информации, ее публикации и распространения. Почти все известные формы представления доступны для современных компьютеров: видео, голос, запахи, тактильные ощущения и многое другое. Подчеркнем, что человек может создавать любую информацию, но для обмена ею с другими людьми надо представить ее в доступной для других и для компьютера форме.

Стоит отметить, что первый и пятый пункты приведенного перечисления подразумевают взаимодействие с внешней средой. А процессы передачи и хранения информации тесно связаны между собой — существует математическая модель, которая обобщает и показывает единство процессов хранения и передачи информации. Основной принцип этой модели заключается в том, что выделяют три главные сущности: отправитель информации, время передачи информации и получатель информации. Если время передачи информации большое и отправитель с получателем информации совпадают, то речь идет о процессе «хранение информации». Если время передачи малое, а получатель и отправитель разные — это процесс «передача информации». Тогда в зависимости от того, кто, кому и за какое время передает информацию, единообразно описываются процессы и хранения информации, и передачи информации, и другие возможные их комбинации.

Смысл информатизации в общем представлении направлен на управление разнообразными процессами. По сути любое управление — это достижение какой-либо цели, требуемого состояния по установленным алгоритмам в конкретных внешних условиях и с учетом неизбежных ограничений. Фактически для формирования любого управления может быть создана модель на основе параметров и алгоритмов. В решении задач управления современные компьютеры могут оказать значительную помощь, но, к сожалению, не во всех областях. Если задачу управления можно строго описать, создать четкие алгоритмы и задать пороговые значения всех параметров, то компьютерная система даже без участия человека сможет отлично выполнять функции управления. В настоящее время это актуально для робототехники, особенно для создания автономных интеллектуальных мобильных роботов.

С учетом вышесказанного можно сделать вывод о том, что самыми важными для внутреннего развития ИИ являются два основных действия: обработка и представление информации, и их развитие позволит существенно усовершенствовать системы ИИ. В учебном пособии рассматриваются новая технология эволюционного накопления информации в миварном информационном пространстве и новая технология логико-вычислительной обработки информации на миварных сетях с линейной сложностью логического вывода, которые и образуют основу миварного подхода. Отметим, что миварный подход изначально был создан для создания глобальных познающе-диагностических систем [1].

1.3. Познающе-диагностические автоматизированные информационные системы и сложные предметные области

Для выявления предельных требований к автоматизированным системам необходимо описать максимально сложные условия их применения. Выделяют классы познающих и диагностических автоматизированных систем обработки информации (АСОИ):

- основной задачей познающих систем является изучение новой сложной предметной области без существенных ограничений по времени работы;
- основной задачей диагностических систем является другая крайность — в короткие сроки принимать решения в динамической и хорошо изученной (формализованной) предметной области.

Представляется, что наиболее сложным случаем можно назвать сочетание этих двух систем, когда в неизвестной исследуемой предметной области надо распознавать ее состояние и принимать решения в короткие сроки. В подобных познающе-диагностических АСОИ отрабатываются новые подходы, модели, методы и алгоритмы. Практически все реальные сложные проблемы менеджмента относятся именно к познающе-диагностическим задачам. Таким образом, для обоснованной классификации АСОИ целесообразно сформулировать наиболее важные условия и категории сложности различных предметных областей. Все управленческие АСОИ и программы можно будет разбить на классы решаемых задач по сложности. Тогда разные пользователи смогут обоснованно выбирать для себя наиболее подходящие конкретные АСОИ.

С научной точки зрения наибольший интерес представляют максимально сложные предметные области. Ведь если некая АСОИ создана для наиболее сложного случая, то она сможет решать и более простые задачи даже в упрощенном варианте самой АСОИ. Кроме того, с помощью исследования максимально сложных случаев можно выявить предельные возможности современных АСОИ, сформулировать научные проблемы и приступить к их целенаправленному решению.

Наиболее сложными считаются следующие условия для системы управления:

- 1) масштабный, разнообразный, изменяющийся и развивающийся объект управления, когда принципиально нельзя сделать его полную информационную модель;
- 2) объективное наличие и сильное влияние фактора случайности событий, их непредсказуемость;
- 3) агрессивная внешняя среда с частыми, неожиданными и очень быстрыми изменениями;
- 4) ограниченные внешние и внутренние ресурсы, которых не хватает для всех, что порождает конфликты и конкуренцию;
- 5) наличие не менее интеллектуальных и не менее сильных объектов-противников или конкурентов;
- 6) проблемы со своевременностью получения и передачи сигналов управления: длительные задержки при передаче сигналов управления и получении сигналов с датчиков (в пределе счет идет на доли секунды);
- 7) проблемы с полнотой исходных данных (все данные получить невозможно);
- 8) проблемы с достоверностью получаемых исходных данных;
- 9) важность и многоаспектность принимаемых решений.

Стоит отметить, что в таких случаях принципиально невозможно разработать идеальную систему управления (не хватает либо ресурсов, либо времени). Как правило, существует несколько вариантов создания подходящих систем управления, из которых надо выбрать оптимальный. Важно подчеркнуть, что на выходе получают квазиоптимальную систему, а так как внешняя среда и условия постоянно изменяются, то и эта система должна быть открытой и эволюционной.

Важно еще и то, что, когда некие действия уже начались, у менеджеров (лиц, принимающих решения, — ЛПР) нет времени на раздумывание и создание новых планов действий, им остается только выбрать какой-то заранее разработанный план и реализо-

вывать его с осознанием всей ответственности и, возможно, немного модернизировать и уточнять его по ходу совершения последовательности действий и получения обратной связи о развитии ситуации. Для таких сверхсложных случаев существующие традиционные базы данных и простейшие экспертные системы не могут быть адекватным решением.

Именно для подобных ситуаций, для создания познающе-диагностических систем, функционирующих в режиме реального времени, и применяются миварные технологии, разрабатывались миварные базы данных и правил на основе гносеологического информационного пространства «Вещь–Свойство–Отношение» [1–4, 13]. Стоит обратить внимание на то, что в таком пространстве возможно параллельное создание нескольких информационных моделей, сопоставление их результатов и формирование различных прогнозов. Это вполне соответствует современным технологиям: сервисно-ориентированным архитектурам, «облачным» вычислениям, многоагентным системам.

1.4. Технологии и достижения ИИ

Интеллект — это сложная область знаний, которую невозможно описать с помощью какой-то одной теории. Ученые строят целую иерархию теорий, характеризующих его на разных уровнях абстракции (уровни исследований в области ИИ).

1. «Рефлексный уровень» — на самом низком уровне находятся нейронные сети, генетические алгоритмы и другие формы эволюционирующих вычислений, позволяющие понять процессы адаптации, восприятия, воплощения и взаимодействия с физическим миром (нередко этот уровень называют «до-интеллектуальным» или рефлексным).
2. «Логический уровень» — на втором уровне располагаются более строгие шаблоны логического вывода, а ученые исследуют схемы дедукции, абдукции, индукции, поддержки

истинности и другие бесчисленные модели и принципы рассуждений. Также на этом уровне разрабатывают экспертные системы, интеллектуальные пакеты прикладных программ (ИППП) и системы поддержки принятия решений. Этот уровень часто называют «интеллектуальным».

3. «Социальный уровень» — на третьем уровне абстракции находятся системы полноценного понимания естественного языка, эмоций и т.п., а также пытаются моделировать роль социальных процессов в создании, передаче и подкреплении знаний. Такие исследования относятся к социальному «сверхинтеллектуальному» уровню.

Стоит отметить, что рассмотренный в учебном пособии миварный подход предназначен прежде всего для работы на втором уровне, который расширен с логической обработки до обработки информации с учетом автоматического конструирования алгоритмов. Кроме того, опыт исследований показал, что на этом уровне необходимо учитывать и традиционные вычислительные процедуры и подпрограммы.

Необходимо напомнить, что согласно принятой трактовке искусственный интеллект предназначен для автоматизации разумной деятельности. Он опирается на теоретические и прикладные принципы информатики: структуры данных, алгоритмы, языки и методики программирования. Отсутствие общепринятой дефиниции для термина «искусственный интеллект» приводит к потребности в создании шкалы, определяющей уровень интеллектуальности, о чем уже упоминалось в начале этой главы.

Не надо забывать, что до появления человека природа прошла длинный эволюционный путь. Необходимо разобраться с интеллектуальностью на уровне животных, что уже будет весьма полезно для промышленности, а потом браться за создание искусственного человека. Для многих практических задач достаточно моделирования и реализации деятельности даже простейших животных (на «рефлексном» уровне).

Необходимо отдать должное языку Фреге, изложенному в 1884 г. и именуемому сейчас исчислением предикатов первого порядка [17]. Он служит инструментом для записи теорем и задания значений истинности, а также сыграл принципиальную роль в создании теории представления для ИИ. Исчисление предикатов первого порядка обеспечивает средства автоматизации рассуждений: язык для построения выражений; теория, позволяющая судить об их смысле; логически безупречное исчисление для вывода новых истинных выражений.

Помимо этого, для фундаментальных принципов ИИ особенно важна работа Рассела и Уайтхеда [20], в которой используется подход к математике как к чисто формальной системе. Это означает, что аксиомы и теоремы рассматривались только в качестве наборов символов, а доказательства должны были выводиться лишь посредством применения формальных правил. При этом исключалось использование интуиции и не требовалось понимание смысла теорем для их доказательства. Смысл, содержащийся в теоремах и аксиомах системы, в данном подходе совершенно не зависит от логического вывода. Такой формальный подход к математическим умозаключениям сделал возможной его автоматизацию в реальных вычислительных машинах.

Логический синтаксис и формальные правила вывода, разработанные Расселом и Уайтхедом, лежат в основе систем автоматического доказательства теорем и составляют основы искусственного интеллекта. Однако в то же время их исследования наложили важные ограничения на принципы описания реальных предметных областей.

Стоит также отметить А. Тарского [19], чья теория ссылок сыграла принципиальную роль в процессе формирования ИИ. Согласно этой теории правильно построенные формулы Фреге или Рассела–Уайтхеда определенным образом ссылаются на объекты реального мира. Эта концепция лежит в основе большинства теорий формальной семантики [10].

Искусственный интеллект не был жизнеспособной научной дисциплиной до появления цифровых вычислительных машин. Однако архитектура цифровых компьютеров наталкивает на специфичное представление теории ИИ, на представление об интеллекте как о способе обработки информации. Дж. Люгер [10] формулирует интересную мысль, с которой невозможно не согласиться: «Мы часто забываем, что инструменты, которые мы создаем для своих целей, влияют своим устройством и ограничениями на формирование наших представлений о мире». Такое, казалось бы, стесняющее кругозор взаимодействие является важным аспектом развития человеческого знания — инструмент (а научные теории в конечном счете тоже инструменты) создается для решения конкретной проблемы. По мере применения и совершенствования инструмент подсказывает другие способы его использования, которые приводят к новым вопросам и, в конце концов, к разработке новых инструментов.

Кроме того, стоит отметить наличие биологических и социальных моделей интеллекта, многоагентных систем, которые опираются не на исчисление предикатов, а на совсем другие и гораздо более современные научные теории. Также существует интересное определение термина «интеллект», согласно которому интеллект заключался не в знании истины, а в знании, как вести себя в постоянно меняющемся и развивающемся мире. Таким образом, интеллект рассматривается с точки зрения выживания в мире как набор логических утверждений о мире (в сочетании со схемой вывода).

Из описания различных моделей интеллекта можно выделить несколько наиболее важных и актуальных для миварного подхода и его приложений. В нейронных моделях [6] искусственного интеллекта упор делается на способность мозга адаптироваться к миру, в котором он существует, с помощью изменений связей между отдельными нейронами. Знание в этих системах не выражается логическими конструкциями, а пред-

ставляется в неявной форме как свойство конфигураций таких связей.

Другая модель интеллекта, заимствованная из биологии, навеяна процессами адаптации видов к окружающей среде. В разработках генетических алгоритмов программы не решают задачи посредством логических рассуждений, а порождают популяции соревнующихся между собой решений-кандидатов и заставляют их совершенствоваться на основе эволюции.

Социальные системы дают еще одно модельное представление интеллекта с помощью глобального поведения, а не поведения отдельных членов общества. В основе такого подхода лежат два аспекта:

- 1) корни интеллекта связаны с культурой и обществом, а следовательно, разум является эмерджентным;
- 2) разумное поведение формируется совместными действиями большого числа очень простых взаимодействующих полуавтономных индивидуумов или агентов.

Именно взаимодействие агентов создает интеллект. Агентом можно считать элемент сообщества, который может воспринимать аспекты своего окружения и взаимодействовать с этой окружающей средой либо непосредственно, либо путем сотрудничества с другими агентами. Большинство интеллектуальных методов решений практических задач требует наличия разнообразных агентов. Это могут быть простые агенты-механизмы, задача которых собирать и передавать информацию; агенты-координаторы, которые обеспечивают взаимодействие между другими агентами; агенты поиска, которые перебирают пакеты информации и возвращают какие-то избранные частицы; обучающие агенты, которые на основе полученной информации формируют обобщающие концепции; и принимающие решения агенты, которые раздают задания и делают выводы на основе ограниченной информации.

Получается, что в рамках данной концепции механизмы обеспечивают выработку решения в условиях ограниченных ресурсов и процессорных мощностей. Для разработки и построения таких агентских сообществ необходимы структуры для представления информации, стратегии поиска в пространстве альтернативных решений и архитектура, обеспечивающая взаимодействие агентов. Технология многоагентных систем уже фактически использована для создания познающе-диагностических систем, хотя подобные описания и термин «агенты» используются не всегда.

Таким образом, важно подчеркнуть, что в современной теории ИИ кроме предикатного подхода уже разработано большое количество других, кардинально отличающихся от него, подходов. Одним из них является продукционный подход, который нашел отражение в миварных сетях.

Как уже упоминалось, в области ИИ существует две фундаментальные проблемы — представление знаний и поиск. Первая проблема относится к получению новых знаний с помощью формального языка. Вторая проблема (поиск) — это метод решения проблемы, в котором систематически просматривается пространство состояний задачи, т.е. ее решения. Эти проблемы являются камнем преткновения для создания систем искусственного интеллекта и прослеживаются с самых истоков.

Одной из первых тем исследований в области ИИ были игровые интеллектуальные задачи: игры, такие как пятнашки, шашки, шахматы и т.п. Эти игры ведутся с использованием четко определенного набора правил, что уже само по себе является важным ограничением по сравнению с реальными задачами, где правил может не быть совсем или они могут изменяться по желанию противников. Позиции фигур легко представимы в компьютерной программе и не требуют создания сложных формализмов, необходимых для передачи семантических тонкостей более сложных предметных областей, а тестирование игровых программ не порождает никаких финансовых или этических проблем.

Поиск в пространстве состояний — принцип, лежащий в основе большинства исследований в области ведения игр. Игры могут порождать большие пространства состояний, а для поиска в них требуются мощные методики, которые называют «эвристиками». Они имеют положительные и отрицательные стороны. Плюсы в том, что эвристики ускоряют поиск и определяют, какие альтернативы следует рассматривать в первую очередь, что значительно сокращает пространство состояний задачи. Минусы в том, что они потенциально способны упустить правильное решение. Большая часть того, что называется разумностью, опирается на эвристики, которые люди используют в решении задач. Отмечено, что наличие противника усложняет структуру программы, добавляя в нее элементы непредсказуемости.

Как видно из вышесказанного, в XX веке в области ведения игр решались сравнительно простые задачи, и в настоящее время такое направление является скорее коммерческим, так как в научном плане используемые в данной области подходы являются «тупиковыми».

Следующей активно развивающейся областью ИИ стало автоматическое доказательство теорем. Эта область является одной из старейших в ИИ, и здесь было много достижений, исследований и программ, включая Универсальный решатель задач (GPS) Ньюэлла и Саймона [18]. В данной области были формализованы алгоритмы поиска и разработаны языки формальных представлений, такие как исчисление предикатов и логический язык программирования Prolog. Привлекательность автоматического доказательства теорем основана на строгости и общности логики. Разнообразные задачи можно попытаться решить, если представить описание задачи и относящуюся к ней информацию в виде логических аксиом и рассматривать различные случаи задачи как теоремы, которые нужно доказать. Однако, несмотря на привлекательность данного подхода, систему, которая единообразно решала бы сложные задачи, разработать так и не удалось. Это натолкнуло ученых на необходимость создания

принципиально новых подходов, к числу которых относится и миварный.

Приведем обоснование ограниченности возможностей автоматического доказательства теорем, которое имеет большое значение для дальнейших исследований и обрисовывает «тупиковые» пути развития.

Любая относительно сложная логическая система способна сгенерировать бесконечное количество доказуемых теорем. Без мощных методик (эвристик), которые бы направляли поиск, программы доказывали большие количества не относящихся к делу теорем, пока не натыкались на нужную. Из-за этой неэффективности многие утверждают, что чисто формальные синтаксические методы управления поиском в принципе не способны справиться с такими большими пространствами, и единственная альтернатива этому — положиться на неформальные, специально подобранные к случаю (лат. «ad hoc») стратегии, как это делают люди. Это один из подходов, лежащих в основе экспертных систем, и он оказался достаточно плодотворным.

Таким образом возникают новые проблемы — работа с бесконечными множествами теорем и создание эвристик (про которые ранее было сказано, что они не гарантируют решение задачи). Стоит отметить, что эвристики предназначены для изменения порядка перебора вариантов решений, но в «худших случаях» не уменьшают общее количество вариантов и такие задачи «по классике» остаются полно-переборными (NP-полными). Конечно, за прошедшее время ученым удалось разработать мощные эвристики, основанные на оценке синтаксической формы логического выражения, которые в результате помогают в среднем быстрее находить правильное решение и «понижают сложность» пространства поиска. Кроме того, пришло понимание, что системе не обязательно решать особо сложные проблемы без человеческого вмешательства. Многие современные программы доказательств работают как умные помощ-

ники, предоставляют людям разбивать задачи на подзадачи и продумывать эвристики для перебора в пространстве возможных обоснований. Этот вывод может служить подтверждением тезиса о том, что ИИ — усилитель человеческих способностей и автоматизация (материализация) мыслительных процессов. Помимо этого, он косвенно подтверждает необходимость введения шкалы измерений интеллектуальности автоматических систем и наличие относительно слабых форм интеллекта у уже существующих компьютерных программ и устройств. Стоит отметить, что существуют подходы в «измерении интеллекта» по объему, структурированности, сложности, многомерности информации и по количеству одновременно обрабатываемых правил в решении задач принятия решений.

Вместе с тем рассмотренные проблемы доказательств поднимают вопросы о необходимости формализации представлений и описаний различных предметных областей. Однако такая возможность есть далеко не всегда. Следовательно, надо изучать возможности обработки менее формализованной информации

или альтернативных моделей представления данных и знаний, что также учитывается в миварном подходе. Он позволяет работать с различными формами представления данных и правил (знаний), включая бесконечные описания сущностей — вещей, отношений и свойств в миварном многомерном динамическом информационном пространстве унифицированного представления данных и правил (знаний).

Следующее направление ИИ — создание экспертных систем. Среди главных достижений ранних исследований по ИИ стало осознание важности специфичного для предметной области знания. Экспертное знание — это сочетание теоретического понимания проблемы и набора эвристических правил для ее решения. Экспертные системы (ЭС) создаются с помощью заимствования знаний у эксперта и кодирования их в форму, которую компьютер может использовать для решения аналогичных

проблем [8]. Следовательно, стратегии ЭС основаны на знаниях человека. К настоящему моменту разработаны сотни ЭС, которые прекрасно зарекомендовали себя в различных предметных областях для решения относительно сложных задач. Современные ЭС могут работать на основе ненадежной или недостаточной информации, выводят ясные и логичные пояснения своих решений. Однако большинство ЭС были написаны для специализированных предметных областей, которые довольно хорошо изучены и располагают четко определенными стратегиями принятия решений, но многие проблемы реализовать в ЭС гораздо сложнее. Принято выделять пять основных проблем возникающих в ЭС:

- 1) трудности в передаче глубоких знаний предметной области и потенциальная ограниченность знаний ЭС;
- 2) недостаток здравого смысла и гибкости. ЭС не могут подобно людям исследовать сначала общие принципы, а потом вырабатывать стратегию для подхода к проблеме;
- 3) неспособность давать осмысленные объяснения. Поскольку ЭС не владеют глубоким знанием своей предметной области, их пояснения обычно ограничиваются описанием шагов, которые система предприняла в поиске решения, но они не могут объяснить, почему был выбран конкретный подход;
- 4) трудности в тестировании. Обосновать корректность работы любой большой компьютерной системы — достаточно трудоемкая задача. Проверять ЭС особенно тяжело и корректность их работы — это серьезная проблема, так как технологии ЭС применяются для разных критичных задач;
- 5) ограниченные возможности обучения на опыте. ЭС делаются «вручную» и их производительность возрастает только после вмешательства программистов. Это заставляет усомниться в разумности таких систем.

Тем не менее, в настоящее время ЭС нашли самое широкое применение в специализированных областях, а теория ЭС продолжает развиваться. Одним из путей развития ЭС является их разработка на основе миварного подхода, который позволяет устранить указанные выше ограничения и выйти на новый уровень интеллектуальности ЭС. Стоит отметить, что именно широкое использование ЭС дает надежные основания для утверждения, что ИИ уже создан и активно применяется людьми в повседневной деятельности.

Нельзя не отметить следующую популярную тему исследований в области ИИ — понимание естественных языков и семантическое моделирование. Способность применять и понимать естественный язык является фундаментальным аспектом человеческого интеллекта. Успешная автоматизация такой способности привела бы к огромному повышению эффективности самих компьютеров. Много усилий потрачено на написание программ, понимающих естественный язык, и эти программы хотя и достигли успеха в ограниченных контекстах, но системы, использующие натуральные языки с гибкостью и общностью, характерной для человеческой речи, лежат за пределами сегодняшних методологий. Известные научные подходы к проблеме понимания естественного языка даже не рассматривают эту проблему в полном объеме, сразу ограничивая область и свои возможности. То есть на текущий момент даже подходов к общему решению этой проблемы нет.

Нельзя не согласиться с тем, что понимание естественного языка подразумевает куда больше, чем разбиение предложений на части речи и поиск значений слов в словаре. Оно базируется на обширном фоновом знании о предмете беседы и идиомах, используемых в этой области, так же как и на способности применять общее контекстуальное знание для понимания недомолвок и неясностей, присущих человеческой речи. Задача сбора и организации этого фонового знания составляет значительную проблему в автоматизации понимания естественного

языка. Разработано множество методов структурирования семантических значений. Но из-за огромных объемов, требуемых для понимания естественного языка, эти методы применяются только в ограниченных предметных областях.

Для передачи семантической организации более богатых и сложных предметных областей ученые ищут формализмы представления, которые должны быть достаточно общими, чтобы применяться в широком круге приложений и уметь адаптироваться к специфичной структуре заданной области. Предложено множество разнообразных методик, большинство из которых являются развитием или модификацией семантических сетей — фундаментальной методики представления семантического значения путем явного выражения отношений связями графа, которые используются при разработке программ, способных понимать естественный язык в ограниченных, но достаточно интересных предметных областях. Кроме того, существуют также стохастические модели, описывающие совместное использование слов в языке, которые применяются для характеристики как синтаксиса, так и семантики. Но полное понимание языка на вычислительной основе все же остается далеко за пределами современных возможностей.

Проблема естественного языка объективно является сложной, более того, вполне возможно, что его вообще нельзя формализовать, и надо использовать принципиально другие подходы. Вместо этого постоянно делаются попытки формализовать естественный язык, которые порождают в ограниченных контекстах узкоспециализированные описания некоторых предметных областей.

Здесь будет уместно привести аналогию с животными. Дрессированные собаки выполняют множество команд своего хозяина, однако эти команды многие хозяева подают не только голосом, но и жестами, свистками и т.п. Получается вполне конкретный набор команд, которые собака может распознать и выполнить.

Нечто подобное может выполнять в ограниченных контекстах узкоспециализированная программа для узких предметных областей. Так и указанный выше подход направлен не на понимание естественного языка, а на понимание команд, т.е. формализованного языка.

Для понимания человеческого языка надо исследовать совсем другие научные подходы, которые принципиально отличаются от современных семантических сетей, онтологий и других формализмов.

Если продолжить аналогию с животным миром, то стоит отметить, что достаточно многие его представители явно обладают интеллектом и умеют решать различные задачи: поиск пищи, распознавание образов, взаимодействие, нахождение пути домой и т.п. Однако человеческий язык во всей своей сложности используется только людьми.

Из вышесказанного можно сделать вывод о том, что понимание естественного языка — это задача такой же сложности, что и создание искусственного интеллекта, равного или превышающего человеческий. Поэтому упрощенные методы никогда не приведут к ее решению, а значит надо разрабатывать новые.

Для понимания языка надо собрать и поддерживать в актуальном состоянии огромную базу данных, фактов и такое же большое количество правил, которые позволяют выявлять нюансы смысла разных понятий в различных ситуациях. Такие возможности на настоящий момент есть у миварного подхода. Покажем это на аналогии: весь язык представляет собой огромный горный массив, вершинами которого являются отдельные слова (рис. 1.1).

Тогда слово — это вершина горы, а гора является контекстом. В процессе обучения человек «выращивает» эти горы у себя в голове, а общается потом только словами, как бы перепрыгивая с вершины на вершину или связывая эти вершины огромными длинными виртуальными мостами. В процессе взросления чело-

век создает целые системы таких гор. Высота каждой горы превышает, допустим, стоэтажный дом. Эти этажи образно соответствуют уровням абстракции в описаниях слов и языковых моделей. Для того чтобы восстановить в компьютере такую горную систему, надо сделать ее полноразмерный математический макет.



Рисунок 1.1. Вид с Эльбруса на Главный Кавказский хребет. Высота 4800 м

Его можно делать, если спускаться этаж за этажом, с вершины горы до ее основания и переходить на другую гору через «долину». Современные методы семантической обработки позволяют сделать, условно говоря, только двух- или трехэтажную по высоте модель смысловой языковой системы.

Возможно, что для полномасштабной системы понадобится работа с бесконечными плоскостями описания сущностей. Именно на это рассчитана идея *миварного подхода* с многомер-

ным информационным пространством и динамически изменяющейся структурой.

Реализация языковых моделей на больших вычислительных кластерах (или ГРИД-системах) на основе миварного подхода должна приблизить нас к созданию автоматической системы, понимающей и разговаривающей с людьми на естественном языке.

1.5. Представление знаний в ИИ

Существует гипотеза о физической символьной системе Ньюэлла и Саймона [18], из которой следует, что интеллектуальная деятельность как человека, так и машины осуществляется на основе следующих средств:

- 1) символьных шаблонов, предназначенных для описания важнейших аспектов области определения задачи;
- 2) операций с этими шаблонами, позволяющих генерировать потенциальные решения проблем;
- 3) поиска в целях выбора решения из числа всех возможных.

Эта гипотеза лежит в основе попыток создания умных машин и неявно различает понятия шаблонов, сформированных путем упорядочивания символов, и среды, в которой они реализованы. Если уровень интеллекта определяется исключительно структурой системы символов, то любая среда, которая успешно реализует правильные шаблоны и процессы, достигнет высокого уровня интеллекта, независимо от того, составлена ли она из нейронов, логических цепей, или чего-то еще. Согласно тезису Черча о вычислимости по Тьюрингу компьютеры способны осуществить любой эффективно описанный процесс обработки символьной информации. Получается, что правильно запрограммированный компьютер обладает интеллектом. Стоит согла-

ситься с гипотезой о физической системе и указать на то, что на основе миварного подхода в скором времени получится реализовать достаточно сложную обработку информации, которую можно будет признать интеллектуальной.

Важно и то, что, по мнению Дж. Люгера, в этой гипотезе указаны главные проблемы исследований в области ИИ:

- представление знаний, т.е. определение структур, символов и операций, необходимых для интеллектуального решения задачи;
- поиск, т.е. разработка стратегий для эффективного и правильного поиска потенциальных решений, сгенерированных этими структурами и операциями.

Необходимо отметить, что не все ученые согласны с указанной гипотезой. Например Т. Виноград и Ф. Флорес утверждают [21], что интеллект является наследственно биологическим и экзистенциальным и не может быть зафиксирован с помощью символов. Подобная критика характерна для таких направлений исследований в области ИИ, как развитие нейронных сетей, генетических алгоритмов и агентно-ориентированных методов. Но здесь еще раз стоит подчеркнуть, что большинство исследований все еще ведется в области рефлексного «доинтеллектуального» уровня ИИ. Видимо, ученые, занимающиеся этими проблемами, именно из-за рефлексной области исследований не соглашаются с гипотезой о физической символической системе. На уровне интеллектуальных исследований преобладает мнение о верности указанной гипотезы.

Вернемся к представлению знаний. Задача любой схемы представления заключается в том, чтобы зафиксировать специфику области определения задачи и сделать эту информацию доступной для механизма решения проблемы. Язык представления должен позволять программисту выражать знания, необходимые для решения задачи.

Абстрагирование, т.е. представление только той информации, которая необходима для достижения заданной цели, является необходимым средством управления сложными процессами.

Конечные программы должны быть рациональными в вычислительном отношении. Выразительность и эффективность являются взаимосвязанными характеристиками оценки языков представления знаний. Многие достаточно выразительные средства представления информации в одних классах задач совсем неэффективны для использования в других.

Разумный компромисс между эффективностью и выразительностью — сложная задача для разработчиков интеллектуальных систем. По существу, способ представления знания должен обеспечить естественную структуру выражения знания, позволяющую решить проблему. Способ представления должен сделать это знание доступным компьютеру и помочь программисту описать его структуру. И здесь разработка новых представлений в виде миварного информационного подхода является закономерным развитием теории ИИ в XXI в.

Особое внимание необходимо уделить передаче сложных семантических значений. Во многих областях ИИ решение задачи требует использования высокоструктурированных взаимосвязанных знаний. Для описания предмета реального мира необходимо не только перечислить его составные части, но и указать способ соединения и взаимодействия этих частей. Структурное представление предметов используется во многих задачах. Кроме того, семантические отношения необходимы для описания причинных связей между событиями.

В обоих случаях взаимосвязи и взаимоотношения могут быть описаны группой предикатов, но для программиста, имеющего дело со сложными понятиями и стремящегося дать устойчивое описание процессов в программе, необходимо некоторое высокоуровневое представление структуры процесса. Предикатное описание можно представить графически, если использовать для

отображения предикатов, определяющих отношения, дуги или связи графа.

Такое описание, называемое семантической сетью, является фундаментальной методикой представления семантического значения. Поскольку отношения явно выражены связями графа, алгоритм рассуждений о предметной области может строить соответствующие ассоциации просто следуя по связям, что значительно эффективнее, чем утомительный и исчерпывающий поиск в базе данных, содержащей описания на языке предикатов.

Очевидно, принцип адекватности представления знаний очень хорошо работает, при одинаковой выразительности семантических сетей и предикатов эффективность поиска явно лучше для семантических сетей с явным описанием связей. Таким образом, остается согласиться с тем, что теория графов эффективно и естественно выражает сложные семантические знания. Кроме того, она позволяет описывать структурную организацию базы знаний. Семантические сети — это достойная альтернатива исчислению предикатов.

Интеллектуальная система должна знать не только сам предмет, но и знать, что она знает этот предмет. Следовательно, рассуждения на метауровне также имеют огромное значение в области ИИ. Осведомленность о своих знаниях составляет более высокий уровень знаний, называемых метазнаниями и необходимых для проектирования и адекватного описания интеллектуальных систем.

Метазнания важны для способности обучаться на примерах, опыте или понимать инструкции высокого уровня, что и отличает их от программирования. Методы представления знаний, разработанные для программирования задач ИИ, обеспечивают возможность адаптации и модификации, столь необходимую для обучающихся систем, а также формируют основу для других исследований с символьными вычислениями.

1.6. Анализ существующих парадигм и моделей обработки данных

Традиционно выделяют следующие парадигмы и модели обработки данных: исчисление высказываний, исчисление предикатов, продукции, семантические сети, онтологии и др. У продукционного подхода есть важные преимущества. Знания о внешнем мире могут иметь двоякую природу, они могут содержать:

- декларативное описание фактов и явлений внешнего мира, фиксирующее их наличие или отсутствие, а также основные связи и закономерности, в которые эти факты и явления входят;
- процедурные описания того, как надо манипулировать этими фактами и достигать целей, интересных для системы.

Продукции в общем виде записывают в форме «если..., то...». Часть специалистов по интеллектуальным системам считает, что запись знаний в виде систем продукций носит универсальный характер — любые знания можно записать в такой форме.

В системе продукций можно представлять самые разнообразные правила, процедуры, формулы или сервисы. К ним, по сути, сводятся все каузальные, т.е. причинно-следственные, утверждения. Применение продукционного подхода для логико-вычислительной обработки разнообразных данных является вполне обоснованным и целесообразным.

Для решения многих практических задач требуется проводить как логическую, так и вычислительную обработку данных. Исторически сложилось, что области логического вывода и вычислительной обработки развивались самостоятельно и успешно решали различные классы задач. В некотором смысле даже существовало противоречие между этими подходами. Кроме того, разделялись проблемы обработки и хранения информации.

Базы данных преимущественно использовались только для хранения и поиска требуемых сведений, а системы логического вывода и вычислений применялись для обработки информации, поиска решений и т.п. Получалось, что эти области относительно слабо пересекались, хотя в плане перспектив развития в каждой из них регулярно провозглашались цели объединения всех функций по накоплению и обработке информации в одной системе.

Для анализа проблем метода логико-вычислительной обработки данных и его возможных применений очень важным является то, что ядром всех основных типов рассмотренных интеллектуальных систем являются база знаний и блок, осуществляющий вывод с помощью знаний (решатель, «движок», «разуматор», планировщик или логический блок) либо выполняющий автоматическое конструирование алгоритма решения из разных модулей и подпрограмм. Этот логический вывод (конструирование алгоритмов) составляет основную процедуру, реализуемую в интеллектуальных системах.

В настоящее время продолжается дискуссия о роли и возможных применениях различных логических механизмов и прежде всего исчисления высказываний, предикатов и продукций. У продукционного подхода есть важные преимущества. Для описания знаний в интеллектуальных системах используются специальные языки описания знаний (ЯОЗ). Простейшими видами таких ЯОЗ являются языки исчисления высказываний или исчисления предикатов вместе с теми процедурами вывода, которые для них известны. Хотя в современных интеллектуальных системах подобные языки используются довольно редко. Куда более распространены в них языки, основанные на продукциях. Продукции в общем виде можно записать в форме «если..., то...», однако к продукциям относятся выражения не только имеющие эту форму, но и многие другие. Продукции могут иметь весьма различное значение. В качестве их левых и правых частей могут выступать и некоторые утверждения,

и действия. Также имеется немало примеров, когда знания, внешне не имеющие продукционной формы, удается перевести в систему продукций. К продукциям сводятся все каузальные, т.е. причинно-следственные, утверждения. Стоит отметить, что под продукциями можно рассматривать множество правил вывода в канонических системах (системах продукций Поста), в которых есть посылки и следствия. С точки зрения анализа миварных сетей и миварного метода логико-вычислительной обработки данных принципиально важным является то, что в системе продукций можно представлять самые разнообразные правила, процедуры, формулы или сервисы. Следовательно, применение продукционного подхода для логико-вычислительной обработки разнообразных данных является целесообразным.

Продукционной системой будем называть любую совокупность продукций, в которую могут входить продукты любого из перечисленных выше типов. Существуют различные конструкции продукций. В наиболее общем виде вместо продукций типа $a \Rightarrow b$ рассматривают более сложные конструкции. В общей форме продукции имеют вид:

$$i, \Pi, P, A \Rightarrow B, Q$$

Здесь $A \Rightarrow B$ — обычная продукция «если..., то...», которая носит название ядра продукции. Элемент P характеризует внешние условия или условия применимости продукции, определяемые факторами, не входящими непосредственно в A , например целями, которые стоят перед рассуждающей системой. Условия P позволяют из всех продукций, у которых в левой части ядра стоит A , отбирать нужную часть продукций. Элемент Π характеризует сферу проблемной области базы знаний или предусловия применимости продукции. Эти предусловия ничем не отличаются от P , но выделяют подсистемы продукций на ранг выше тех, которые выделяют условия. Предусловия задают формальную систему, в рамках которой будут проводиться логи-

ческие рассуждения. Под элементом i понимают уникальный идентификатор продукции. Наконец, Q характеризует постуловия продукции, указывающие на те изменения, которые необходимо внести в базу знаний и в систему продукций после реализации данной продукции. В общем виде продукции встречаются весьма редко.

Под логической обработкой принято понимать некий вывод, лежащий в основе человеческих рассуждений. Для проведения анализа используются описания и исходная информация из широко известных источников. Всякий вывод, вне зависимости от его организации, «по классике» носит переборный характер. Программа вынуждена перебирать варианты, заходить в тупики, проходить циклы прежде, чем она сможет найти правильный путь решения. Повышение эффективности — центральная проблема всех автоматизированных систем дедуктивного вывода. Сразу отметим, что миварные сети позволили выполнять логический вывод без перебора.

Особый интерес представляет описание общей схемы выводов, лежащей в основе большого количества моделей человеческих достоверных рассуждений. Имеется дерево вывода, вершины которого соответствуют определенным утверждением F_i , а дуги определяют порядок получения новых утверждений. Те дуги, которые сходятся в зачерненных точках, образуют *конъюнктивные условия вывода*, а те, которые между собой соединены «дужкой», образуют *дизъюнктивные условия вывода*. Дерево вывода с такими условиями переходов от вершины к вершине носит название *и-или дерева*. В и-или дереве ориентация дуг показывает направление вывода. Естественное разбиение вершин дерева по ярусам отражает глубину вывода (число шагов, необходимых для получения утверждений данного яруса). Первый ярус дерева образуют вершины, играющие роль аксиом или утверждений, истинность которых задается извне. Но схема вывода не обязательно описывается в виде дерева. Она может иметь вид произвольной сети, ориентированной, неориен-

тированной или частично ориентированной. На рисунке 1.2 показан пример неориентированной сети. Такая сеть (наличие или отсутствие ориентации не играет здесь роли) называется и-или сетью. Процесс вывода на и-или сети протекает следующим образом. Пусть мы хотим доказать утверждение F_6 (на рис. 1.2 этому соответствует целевая вершина (выделена двойным контуром)). В качестве априорно доказанного задано утверждение F_1 (ему соответствует начальная вершина, которая на рис. 1.2 заштрихована). Как из F_1 можно получить F_6 ? Если считать, что все связи допускают ориентацию в нужную сторону, то из F_1 можно получить F_3 , затем F_5 и, наконец, F_6 . Но этот путь удалось отыскать, потому что сеть, показанную на рис. 1.2, видна «с высоты птичьего полета». Лабиринт поиска представлен в виде чертежа. Именно это позволяет не делать лишних попыток, не двигаться в ненужную сторону, а идти кратчайшим путем к цели. Стоит отметить, что на рис. 1.2 фактически изображен однодольный граф, т.е. все вершины графа принадлежат одному классу. Этот факт будет очень важен для дальнейшего анализа.

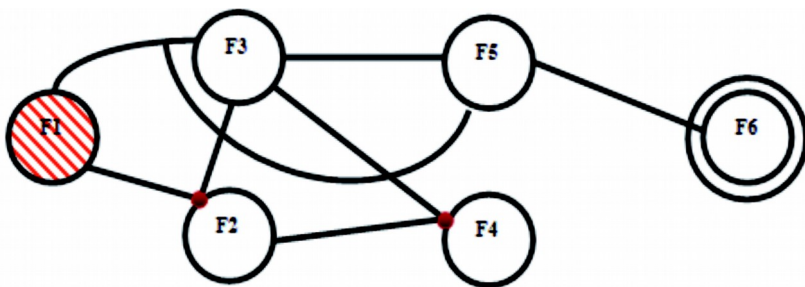


Рисунок 1.2. Пример однодольного графа неориентированной и-или сети

Существуют два метода логического вывода. Один известен под названием *метода прямой волны*. Волна поиска путей к целевой площадке распространяется от всех площадок, играющих роль начальных. Другой — под названием *метода обратной волны*. В этом методе волна начинает свое движение от целевых площадок и движется в направлении начальных площадок лабиринта.

Различие между прямой и обратной волной состоит в том, что они порождают в процессе своего движения различные промежуточные «фронты» площадок, что приводит к различному числу шагов при поиске. Часто используется смешанный метод вывода, при котором одновременно движутся прямая и обратная волны. При встрече этих волн формируется путь вывода от начальных аксиом к целевым выражениям.

Стоит отметить, что этот подход не нашел широкого практического применения в прошлом веке, поскольку не было предложено метода быстрой логической обработки данных. Как было показано выше, ученые исходили из предпосылки, что любой вывод носит переборный характер, что означает факториальный рост вычислительной сложности при увеличении количества правил (NP-полнота). Даже современные суперкомпьютеры при таком подходе могут обрабатывать одновременно в реальном масштабе времени лишь 20 правил. Фактически это позволяет решать только модельные задачи, так как современные требования начинаются с сотен и достигают десятков тысяч переменных и правил даже для относительно простых предметных областей. Миварный подход и создание логико-вычислительных миварных сетей позволили эффективно реализовать предложенные для продукций методы *прямой и обратной волны*.

2. МИВАРНЫЕ ТЕХНОЛОГИИ

2.1. Основы миварного подхода

Как уже было упомянуто выше, из пяти основных процессов, которыми занимается информатика (сбор, передача, накопление, обработка и представление информации) фундаментальными проблемами для внутреннего развития ИИ остаются два основных действия: обработка и накопление информации. Поэтому создание новых подходов для их развития является актуальной задачей.

Одним из таких подходов является *миварный подход*, рассматриваемый в данном учебном пособии. Такой подход позволяет обобщить существующие достижения и предложить новые модели и методы обработки информации и управления ею и базируется на анализе и творческом развитии следующих основных научных направлений:

- философия и гносеология;
- искусственный интеллект;
- кибернетика;
- биология, генетика, математические методы и модели эволюции;
- теории баз данных (БД);
- экспертные системы (ЭС) и базы знаний (БЗ);
- различные модели и методы обработки информации;
- теория графов;
- архитектура ЭВМ;

- распознавание образов;
- психология;
- генетические алгоритмы;
- инновации и др.

Миварный подход включает в себя две основные технологии: накопления и обработки информации.

Миварная технология накопления информации — это способ создания глобальных эволюционных баз данных и правил (знаний) с изменяемой структурой на основе адаптивного дискретного миварного информационного пространства. Эта технология предназначена для хранения любой информации с возможным эволюционным изменением ее структуры и без ограничений по объему и формам представления.

Миварная технология обработки информации — это способ решения логико-вычислительных задач путем создания системы логического вывода или конструирования алгоритмов из модулей, сервисов или процедур на базе активной обучаемой миварной логико-вычислительной сети правил с линейной вычислительной сложностью. Эта технология предназначена для обработки информации, включая логический вывод, вычислительные процедуры и сервисы.

Основой миварного подхода к представлению данных является целостное, адаптивное (эволюционное), динамичное, многомерное описание моделируемой предметной области. Сущность миварной концепции представления данных состоит в следующем. Пусть существует некоторая предметная область и осуществляется ее моделирование. Полученные сведения о предметной области накапливаются в некоторой системе представления данных. Можно ввести следующие три равнозначных и равнозависимых понятия: **вещь** (объект, сущность); **свойство** (атрибут, характеристика); **отношение** (связь, взаимодействие).

Эти три выделенных понятия-категории используются в миварном подходе для представления любой информации о любой предметной области [2]. Стоит отметить, что такие понятия, как, например, элемент, объект, система, функция, структура, организация и подобные им, являются вторичными, производными по отношению к трем основным, а следовательно, могут быть выражены через них. В процессе познания выделяется некоторая вещь в предметной области, а затем ей присваивается некоторое уникальное название. Требование уникальности является существенным, его нарушение может привести к неоднозначностям, а вследствие и к ошибкам. Далее определяются свойства выделенной вещи, на основе которых происходит выделение вещи из предметной области. Каждому свойству, если это не было сделано ранее, присваивается определенное уникальное название (имя, идентификатор). Далее изучаются все отношения выделенной вещи с другими вещами данной предметной области. Каждому отношению присваивается уникальное название (идентификатор). В результате вышеописанного моделирования предметной области образуется множество V названий вещей. Так как каждое название является уникальным, количество элементов этого множества будет равно количеству выделенных вещей в рассматриваемой предметной области. Кроме того, образуется множество S названий свойств. Элементами такого множества являются уникальные названия свойств. Здесь стоит особо отметить, что различным элементам из пространства вещей V , вообще говоря, в зависимости от предметной области, могут соответствовать одинаковые свойства — элементы из пространства S . Элементами множества отношений O являются уникальные названия выявленных отношений вещей в данной предметной области.

Таким образом, в результате моделирования некоторой предметной области образуются три различных множества названий: V , S , O . В рамках миварной концепции представления данных считается, что любая вещь — это некоторое название этой вещи,

некоторая совокупность ее свойств и отношений, в которых эта вещь себя проявляет. В определенных ситуациях вещь может рассматриваться как свойство или как отношение, свойство — как вещь или отношение и отношение как вещь или свойство. Таким образом, эти три понятия являются неразрывными и взаимопроницаемыми и лежат в основе построения картины мира.

Стоит подчеркнуть, что в традиционных концепциях представления данных оперируют прежде всего вещами отдельно. Свойства же и отношения для данной вещи являются понятиями побочными. Примером такой концепции является графотабличная концепция представления данных, получившая наибольшее развитие в модели данных «сущность—связь». В миварном же подходе к представлению данных все эти три понятия равноправны.

Таким образом, на основе трех выделенных множеств V , S и O конструируется *миварное пространство*.

Миварное пространство — это многомерное (трехмерное) дискретное пространство, которое задается путем определения основных осей и фиксации основных объектов предметной области, свойств и отношений.

Мивар — это наименьший элемент миварного пространства.

Такая модель представления данных находит отражение во многих областях искусственного интеллекта и может быть использована для моделирования предметной области при создании экспертных систем, систем поддержки принятия решений (СППР), автоматизированных систем управления (АСУ), роботов, а также систем распознавания образов, поисковых систем, систем смыслового понимания текста, аннотирования и перевода текста и др.

Следующим базовым понятием рассматриваемого миварного подхода является понятие *миварной сети* (МС). Именно

миварная сеть обеспечивает формализацию и представление человеческих знаний при создании экспертных систем.

Миварная сеть — это способ представления части информации миварного пространства (объектов и правил их обработки) в виде двудольного ориентированного графа [1, 3, 4], состоящего из объектов P и правил R , где $P \subseteq V$, а $R \subseteq O$.

Напомним определение двудольных графов. Граф $G = (V, E)$ называется двудольным, если существует разбиение $V = \{V_1, V_2\}$ такое, что никакие две вершины из V_1 или из V_2 не являются смежными. Введенные понятия допускают естественное обобщение. Неориентированный граф называется k -дольным, если его вершины можно разбить на k классов так, что концы каждого ребра принадлежат разным классам. Таким образом, можно использовать и двудольные, и трехдольные и многодольные (k -дольные) графы для разных предметных областей.

Следует отметить следующие важные свойства миварной сети.

1. Сеть состоит из элементов двух видов (двух долей графа): вершин графа — объектов P и ребер графа — правил R .
2. Для каждой переменной P в явном виде хранится информация о всех правилах R , для которых она является входной X или выходной Y переменной с явным указанием этого.
3. Для каждого правила R в явном виде хранится информация обо всех его входных и выходных переменных P , включая и информацию о количестве входных X и выходных Y переменных.
4. Хранение всей необходимой информации такой сети организуется на основе технологий баз данных, адаптированных под работу с миварным информационным пространством.
5. В каждом элементе миварной сети, будь то вершина или ребро, однозначно и полностью определены все смежные

ребра и вершины. В любом месте миварной сети всегда известно, откуда в него можно прийти и куда из него можно перейти, что исключает использование переборов при поиске логического вывода на миварной сети. Это является прямым следствием того, что МС представляется в виде двудольного ориентированного графа.

Таблица 2.1. МС в виде матрицы

Правила	Параметры								
	1	2	3	4	5	...	n-2	n-1	n
1	X	X	X					Y	Y
2			X	Y	Y			X	X
...									
m		X		X	X		Y		

Биграф миварной сети может быть представлен в виде двумерной матрицы $P \times R$, где n — число параметров (объектов) предметной области, m — число правил связывающих объекты предметной области, следующим образом (табл. 2.1).

Таким образом, миварная сеть является особым способом формализации и представления информации в виде множеств двух и более типов, которые потом связываются между собой. Множество первого типа называется «Объекты», множество второго типа называется «Правила». Правила в общем виде состоят из набора входов, набора выходов и внутреннего алгоритма обработки входных данных в выходные. Правилами, например, могут быть математические формулы, производственные правила «если..., то...», индикаторы типа связи двух объектов, целые сервисы. Уже два этих множества позволяют построить двудольный ориентированный граф за счет связывания множеств разных типов: «объект–правило» и «правило–объект». Взаимосвязи типа «объект–объект» и «правило–

правило» запрещены. В общем виде взаимосвязь имеет вид: «объект(ы)–правило–объект(ы)». Первым указывается элемент, из которого исходит взаимосвязь. Вторым указывают тот элемент, куда приходит взаимосвязь. За счет этого устанавливается ориентированность графа и исключаются возможности неверного толкования или преобразования объектов за счет обратного прохода по взаимосвязи.

При таком построении МС обладает свойством масштабируемости, так как в любой момент времени можно добавить элементы множеств любых имеющихся типов без необходимости изменения методов их обработки.

Стоит отметить, что одна из ключевых особенностей МС заключается в том, что для ее описания необязательно нужен эксперт. В большинстве случаев достаточно просто перенести существующие объекты и связи (правила) в миварный вид.

То есть миварные сети могут быть представлены в виде двудольного графа, состоящего из объектов-переменных и правил-процедур. Для этого составляются два списка: список объектов (изображены в виде овалов) и список правил (изображены в виде прямоугольников, рис. 2.1). Каждое правило в миварной сети является развитием продукций, метаправил или вычислительных процедур. Кроме того, стоит отметить, что в качестве миварных правил могут быть использованы различные сервисы, модули и вычислительные процедуры. Доказано [1], что с точки зрения дальнейшей обработки все эти формализмы идентичны и представляют собой по сути вершины двудольного графа. На рис. 2.2 показан пример записи объектов и правил миварной сети в формате XML, а алгоритм логико-вычислительной обработки с использованием программного комплекса «Конструктор экспертных систем миварный Wi!Mi РАЗУМАТОР 2.1» показан на рис. 2.3.

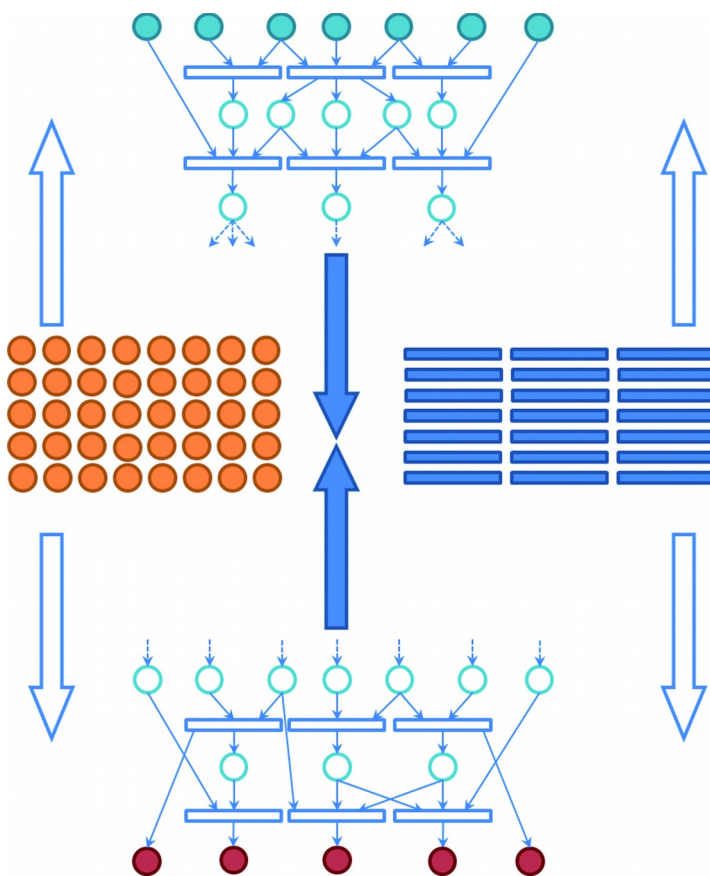


Рисунок 2.1. Формирование двудольной миварной сети

Миварные сети позволяют развить продукционный подход и создать автоматическую обучаемую логически рассуждающую систему реального времени. В работах [1, 3, 4, 22–109] показано, что миварный подход объединяет и развивает продукционные системы, онтологии, семантические сети, сервисно-ориентированные архитектуры, многоагентные системы и другие современные информационные технологии в целях создания интеллектуальных систем и систем ИИ.


```
<?xml version="1.0" encoding="UTF-8" ?>
<-root>
<-parametr>
<parametr id="P1" value="0.0" description="Угол А, против. стороне а (град)" />
<parametr id="P2" value="0.0" description="Угол В, против. стороне b (град)" />
<parametr id="P3" value="0.0" description="Угол С, против. Стороне с (град)" />
...
</parametr>
<-rules>
<rule id="R1" resultId="P1" initId="P2,P3" value="180-P2-P3" description="Сумма" />
<rule id="R2" resultId="P2" initId="P1,P3" value="180-P1-P3" description="Сумма" />
<rule id="R3" resultId="P3" initId="P1,P2" value="180-P1-P2" description="Сумма" />
...
</rules>
<-metadata>
<idParametr inc="33" />
<idRule inc="161" />
</metadata>
</root>
```

Рисунок 2.2. Пример записи объектов и правил в формате XML

Суть миварного подхода заключается в объединении баз данных и систем логико-вычислительной обработки в единые эволюционно развивающиеся системы, позволяющие собрать вместе различные научные разработки на основе сервисно-ориентированных архитектур и технологий интеллектуальных агентов (многоагентных систем). С появлением такого подхода становится реальным собрать воедино такие элементы, как базы данных, базы знаний, экспертные системы, многоагентные системы, сервисно-ориентированные архитектуры и др., и это позволит создать глобальные самоорганизующиеся программно-аппаратные комплексы с эволюционной структурой для познающе-диагностических и экспертных систем, а также систем оперативной диагностики.

Главное свойство миварного подхода в том, что он формирует основу, фундамент для объединения всех существующих и перспективных наработок в этой и во многих других активно развивающихся научных областях. Формализм миварного многомерного эволюционного пространства унифицированного пред-

ставления данных и правил позволяет описать все существующие традиционные модели данных и методы логической и вычислительной обработки информации. Такое описание и создает предпосылки для объединения всех достижений ИИ.

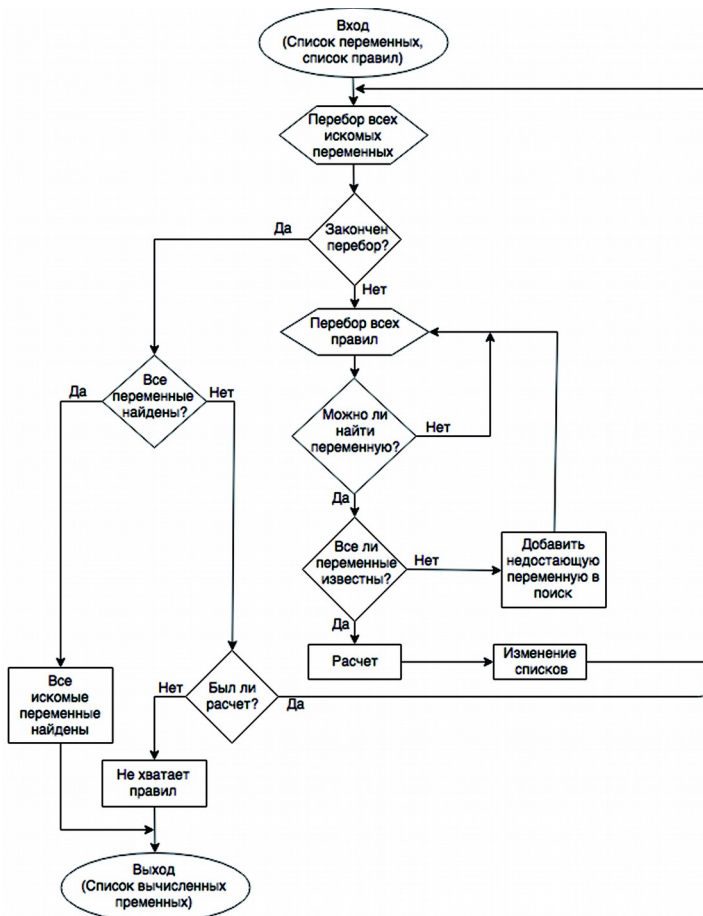


Рисунок 2.3. Алгоритм миварной логико-вычислительной обработки в программном комплексе Конструктор экспертных систем миварный WiMi РАЗУМАТОР

В этом контексте стоит отметить, что миварная модель данных является более мощной, чем известная модель «сущность–связь» (ER-модель), так как она охватывает в своем формализме все традиционные структурные и бесструктурные модели данных, включая семантические сети, онтологии и гипертекст. Реляционные таблицы представляются в многомерном миварном пространстве в виде двумерных таблиц, объединяемых в трехмерное пространство с возможностью дальнейшего наращивания измерений. Сетевые модели, семантические сети и онтологии создаются в виде неких графов, помещенных в дискретное многомерное пространство, что также усиливает возможности подобных сетевых моделей. Более подробно описанное выше представление и возможность перехода из одного представления данных в другое через миварное пространство или одномерные таблицы, которые являются упрощением миварного представления, показано в работе [1]. Необходимо подчеркнуть, что миварное пространство таково, что пользователи могут одновременно оперировать различными моделями данных, начиная с реляционных и гипертекстовых, постепенно вводя все больше структурированности и переходя к сетевым, семантическим сетям и онтологиям, а уже через них к миварному пространству.

В основе миварных сетей лежит продукционный подход («если..., то...»), но с более сложной структурой правил с условиями, условиями, ограничениями, действиями и последствиями. Это позволяет записывать все причинно-следственные отношения, включая и все возможные формы предикатов и логических выражений. Такой подход позволяет работать одновременно с разными описаниями предметных областей, прибавлять к продукциям и предикаты, и нейросети, и генетические алгоритмы, и многоагентные системы, и традиционные вычислительные процедуры, и все другие сервисы с помощью универсальных миварных отношений, которые хранятся перед обработкой в миварном пространстве. Миварные

технологии представления и обработки информации создают многообразие представлений и подходов к решению задач для самых разных предметных областей.

Стоит отметить, что системы, основанные на миварном подходе, очень хорошо распараллеливаются и могут работать в ГРИД-системах, компьютерных сетях и на многопроцессорных вычислительных комплексах. Здесь важно подчеркнуть, что знания в виде процедур, правил и отношений хранятся вместе с фактами в едином миварном пространстве, в результате чего исчезают противоречия между базами данных и базами знаний, так как они гармонично объединены.

Описанная выше технология накопления информации позволяет осуществлять ее эффективную **обработку** в миварных сетях, которые фактически реализуют отношения из классической тройки «вещь–свойство–отношение» [1, 3]. В дальнейшем это позволяет избежать противоречия между хранением и обработкой информации. Кроме того, стоит отметить, что для решения реальных задач важную роль играет форма представления знаний, поскольку разные формы позволяют с разной вычислительной сложностью и достоверностью решать одни и те же задачи. Например, решение логических задач в предметных областях, описываемых более, чем 20 правилами, может привести к NP-полным алгоритмам. Стоит напомнить, что графы и матрицы — это разные формы представления информации (формализмы). Практически любой граф можно представить в виде матрицы и наоборот. Но от формы представления информации очень сильно зависит вычислительная сложность решения разных задач. Миварные сети дают возможность изменить само представление «деревьев решений» (в виде графов), перевести их в матрицы, в том числе многомерные бинарные, и решать задачи, в предметных областях, содержащих миллионы правил и объектов, в реальном времени с линейной вычислительной сложностью. Иногда вывод в таких матрицах называют автоматическим конструированием алгоритмов. Также использование миварных сетей дает возможность получать параллельные решения и подключать сервисы, правила, процедуры, нейропод-

2. Миварные технологии

программы, генетические и т.п. модули, ГРИД-технологии, облачные вычисления, многоагентные технологии.

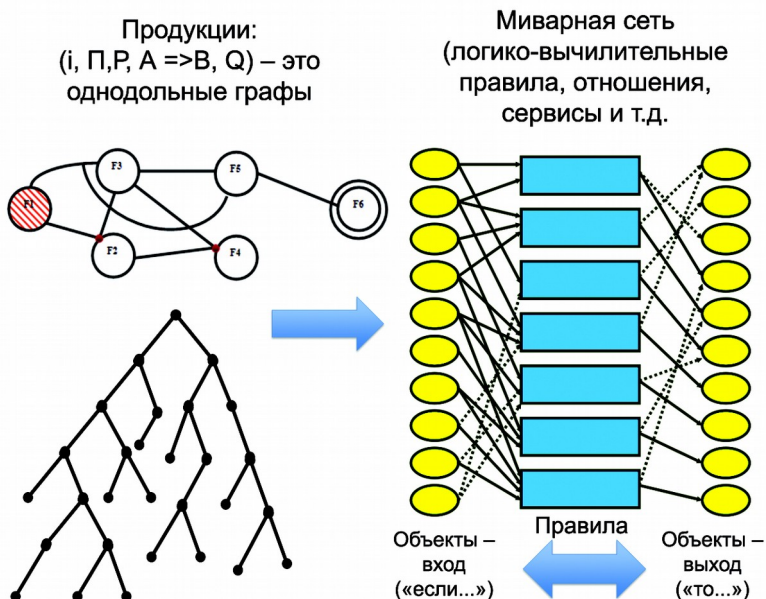


Рисунок 2.4. Переход от продукций к миварным сетям (миварные двудольные графы)

Миварные технологии обеспечивают решение комплексных задач различными методами, включая продукционные, предикатные, нейроматематические, генетические, нечеткие, вероятностные и прочие методы работы как с полной, так и с неполной и противоречивой информацией. Более того, с помощью миварных технологий можно одновременно решать одну задачу разными методами и на разных моделях, что дает возможность выбрать наиболее адекватное решение в заданные сроки и с необходимой точностью.

Таблица 2.2. Формирование двухмерной матрицы миварной сети

Правила	Параметры									
	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	
2			X	Y	Y			X	X	
...						...				
m		X		X	X		Y			
m+1		Z	Z			W	W			

m — количество правил в описании задачи;

n — количество всех объектов-переменных в правилах;

Прямоугольная матрица размером $(m+1; n+1)$ содержит описание предметной области, в которой по строкам расположены все правила, а по столбцам все переменные;

X — входные переменные в конкретном правиле;

Y — выходные переменные в конкретном правиле;

Z — признак известности переменной, который проставляется в $(m+1)$ строке. Это либо заданные входные переменные («Дано»), либо полученные в процессе решения задачи;

W — искомая переменная, значение которой надо найти.

Миварный подход разрабатывается для создания глобальных информационных систем, принципиально открытых для постоянного наполнения новыми данными, с возможностью быстрой обработки и получения требуемых результатов для практических задач. Миварные технологии позволяют гораздо быстрее обрабатывать информацию, например, сводя вычислительную сложность отдельных логических задач при определенных условиях и ограничениях с NP-полной до линейной. На основе данных технологий создан программный комплекс «Конструктор экспертных систем миварный Wi!Mi РАЗУМАТОР», который на практике реализует этот подход. Эта система за счет новой формы представления информации о предметной области способна решать логико-вычислительные

задачи с несколькими миллионами правил и объектов в реальном времени без переборов.

Миварное представление логики позволило перейти от однодольных графов продукций к многодольным сетям и выполнять все операции с линейной сложностью. Иллюстрация такого перехода от однодольных графов продукций («Правила») к двудольным графам миварных сетей («Объекты; Правила») показана на рис. 2.4. Пример формирования двумерной матрицы миварной сети показан в табл. 2.2.

2.2. Почему миварная технология обработки информации быстрее, чем продукции

Миварная технология обработки информации быстрее, чем продукции, потому что:

- 1) в продукциях за основу поиска были взяты правила, которые перебирались для поиска решения, что порождало полный перебор, факториальную сложность и циклы (рис. 2.1);
- 2) в миварных сетях явно выделены две доли: «правила» и «объекты» («переменные»), а за основу поиска алгоритма логического вывода взяты именно «объекты», которые могут иметь только одно значение и их можно найти только один раз, что исключает циклы и полный перебор (рис. 2.1);
- 3) миварная сеть может быть задана двухмерной матрицей (табл. 2.2), в которой каждое правило знает все свои входные и выходные объекты, а каждый объект знает все свои правила и свои роли в них («вход» или «выход»), что позволяет избежать перебора, постепенно выявлять новые известные объекты через соответствующие правила, постоянно сокращать размерность исходной миварной матрицы, обеспечивать линейную вычислительную сложность логического вывода относительно общего количества правил в матрице.

При поиске логического вывода в миварной сети, как показано на рис. 2.1, заданные объекты «вытягивают» соответствующие

им правила, которые затем «вытягивают» соответствующие объекты.

Такое «вытягивание» может быть реализовано и «сверху вниз», и «снизу вверх», и одновременно от «дано» и от «найти». При этом нет никакого перебора, каждое правило и каждый объект используется и ищется только один раз. В худшем случае будет задействована вся логическая сеть путем такого «вытягивания» с линейной вычислительной сложностью относительно количества правил. Если в процессе логического вывода входных данных не будет хватать, то по мере исчерпания всех «запускаемых» правил (т.е. правил, у которых известны все входные объекты) процесс остановится и сообщит о нехватке данных.

В отличие от традиционных подходов, разделяющих хранение в базах данных, логический вывод и вычислительную обработку, миварный подход позволяет создавать многомерные и эволюционные системы, обрабатывающие информацию в реальном масштабе времени с совмещением логических выводов и вычислительной обработки. Основой миварного подхода является то, что реальный мир существует «сам по себе» (без каких-либо абстракций), а при изучении и познании некоторой предметной области человек представляет себе описание этого мира в виде начального трехмерного пространства, осями которого являются понятия «вещь», «свойство» и «отношение». Эти три понятия — абстракции, удобные для описания реального мира.

Миварный подход — это новый подход для разработки интеллектуальных систем и для создания систем логического искусственного интеллекта [1–3].

2.3. Для каких систем создан миварный подход

Множество научных исследований направлено на решение задач с ограничивающими факторами, например, замкнутость описания предметной области, анализируется не более 20–30 логических правил и т.п. Миварный подход создавался для прин-

ципиально других систем, которые можно кратко называть глобальными познающе-диагностическими системами реального времени. Для миварного подхода сразу было поставлено условие работы с открытыми и достаточно большими предметными областями, где счет логическим правилам идет на десятки тысяч.

Глобальные познающе-диагностические системы (ГПДС) должны решать задачу мониторинга и прогнозирования сложной предметной области, в том числе оценку общественно-политической обстановки в реальном времени. Обобщенная схема таких систем ГПДС показана на рис. 2.5.

Стоит отметить, что основой работы глобальных познающе-диагностических систем является информация — глобальная модель предметной области, включающая все данные, все правила, все отношения и т.д. В терминологии миварного подхода это эволюционная база данных и правил. Именно с ее помощью можно по кратким фрагментам получаемых сообщений делать обоснованные выводы и прогнозы. Эти краткие сообщения представляют собой лишь вершины гор (айсбергов), за которыми скрываются огромные описания, хранящиеся в базах данных.

Для решения задачи мониторинга и прогнозирования сложной предметной области необходимо создать автоматизированную систему информационного моделирования и прогнозирования. Прежде всего необходимо разработать информационную модель предметной области и решить задачи сбора требуемых данных, обработки информации и прогнозирования развития объекта наблюдения. Эти классические задачи имеют множество возможных решений, зависящих от особенностей предметной области, имеющихся ресурсов, заданных требований по достоверности, своевременности, надежности, точности прогнозирования и т.п.

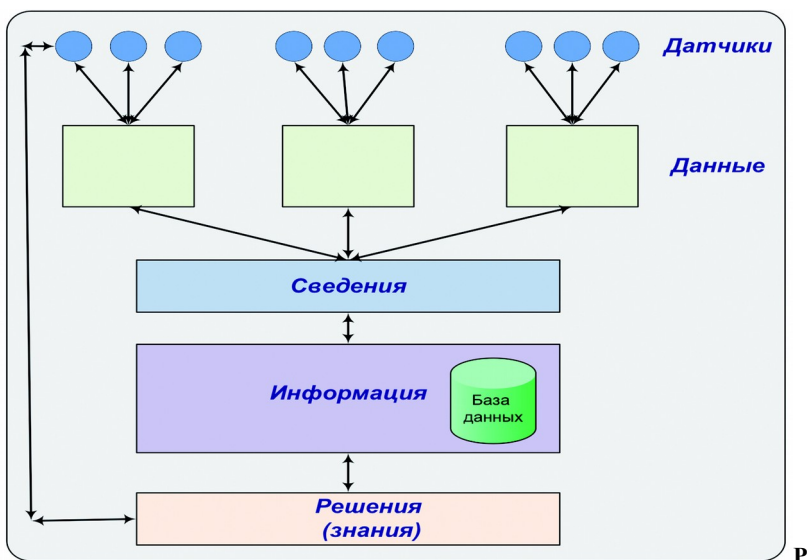


Рисунок 2.5. Система ГИС

В любом случае, необходимо выделение и создание трех подсистем:

- сбора данных;
- накопления и обработки информации;
- прогнозирования и имитационного моделирования.

Создание информационной модели сложной предметной области является творческой задачей с применением информационных технологий баз данных, экспертных систем, систем поддержки принятия решений, интеллектуальных систем и др. Успех решения задачи зависит от концептуального моделирования предметной области (выделяют концептуальный, логический и физический уровни моделирования). Стоит подчеркнуть, что анализ существующих аналогов показывает необходимость изначального применения эволюционных (развивающихся, обучаемых) систем.

В настоящее время есть современные и перспективные методы моделирования самых сложных предметных областей в предельно жестких ограничениях и внешних условиях. К таким информационным технологиям относится, например, отечественная разработка самоорганизующихся программно-аппаратных комплексов оперативной диагностики (СПАКОД) на основе эволюционного многомерного информационного пространства универсального представления данных и правил [1].

При решении задач моделирования и прогнозирования сложных предметных областей невозможно сразу определить требуемые ресурсы и возможные проблемы. Такая задача относится к классу познавательных и не имеет однозначных, а тем более простых, решений. Кроме того, задача мониторинга относится к классу диагностических задач, требующих обработки данных в реальном масштабе времени. Когда задачи познания и диагностики решаются одновременно, возникает наиболее сложный класс познающе-диагностических задач. Сейчас для одновременного решения задач мониторинга и прогноза сложных реальных предметных областей не существует готовых информационных систем.

Однако с помощью комбинирования существующих технических и программных разработок представляется возможным решить требуемую задачу мониторинга и прогнозирования сложных реальных предметных областей.

Например, технология самоорганизующихся программно-аппаратных комплексов оперативной диагностики позволяет постепенно по мере изучения (познания) предметной области эволюционно наращивать требуемые ресурсы.

Информационная модель создается с самого простого и минимального количества данных, а затем по мере поступления новой информации из подсистемы сбора данных происходит эволюционное наращивание информационной модели в подсистеме накопления и обработки данных. Далее появляется

возможность выполнения прогнозирования в различные интервалы времени.

Технология СПАКОД дает возможность одновременно осуществлять информационное моделирование на нескольких моделях, по каждой из которых выполняется прогнозирование. Полученные прогнозы по мере наступления событий сравниваются с реальным развитием ситуации. Затем в рамках организации обратной связи анализируются все прогнозы и их отличие от реального развития событий (ошибки прогнозирования), а также осуществляется модернизация существующих информационных моделей. Практика показывает, что такое разномодельное эволюционное прогнозирование с обратной связью позволяет давать достаточно точные и объективные прогнозы.

Таким образом, технология СПАКОД дает возможность эволюционно и постепенно наращивать как саму информационную модель (и требуемые для ее функционирования ресурсы), так и точность прогнозирования.

Необходимо подчеркнуть, что особенности создания системы сбора данных в основном зависят от поставленных задач и предметной области. Однако технология СПАКОД позволяет в процессе информационного моделирования выявлять недостающие данные, определять, где их можно получить, и управлять процессом сбора данных в реальном масштабе времени. Сбор данных может осуществляться как автоматически в различных компьютерных сетях (Интернет), так и с использованием других технологий, подготовленных людей или роботизированных систем. Известны различные технологии сбора данных, но их целесообразно рассматривать после начального создания информационной модели сложной предметной области.

Если технологии сбора данных сильно зависят от возможностей людей, то следующие этапы (обработка и прогнозирование) позволяют практически полностью автоматизировать весь

процесс и максимально сократить количество требуемого персонала. Кроме того, существуют различные способы разграничения доступа, при которых информация о реально решаемых задачах и получаемых результатах доступна только высшему руководству и системным администраторам СПАКОД.

2.4. Сравнительный анализ миварного подхода и технологий ИИ

Отметим, что миварный подход предназначен прежде всего для работы на втором, интеллектуальном, уровне абстракции, который расширен с логической обработки до обработки информации с учетом автоматического конструирования алгоритмов (более подробно это описано в главе 1). Кроме того, опыт создания интеллектуальных систем показал, что на этом уровне необходимо учитывать традиционные вычислительные процедуры или подпрограммы.

Из наиболее известных формализмов, кроме миварного подхода, этому соответствует подход на основе сервисов или агентов. Есть некая задача, которую формулируют в виде сервиса или агента и затем формализуют в виде «черного ящика» — процедуры с описанием входных и выходных данных. Тогда все процедуры, подпрограммы, агенты, сервисы и т.п. могут быть представлены в виде неких отношений, которые хранятся в базе данных, а при необходимости запускаются на основе использования миварной сети.

Миварный подход предоставляет возможность хранить для одной предметной области несколько разных процедур, решающих одинаковые задачи. Более того, если вычислительные ресурсы позволяют, то все эти процедуры могут запускаться одновременно, а потом на основе конкуренции из представленных ими результатов выбирается наиболее подходящий для каждого конкретного случая.

Важно отметить, что методы первого рефлексного уровня в нашей трактовке могут быть реализованы в виде точно таких

же миварных процедур, включающих и нейронные сети, и генетические алгоритмы, и пр.

Также перечисленные выше формализмы Фреге, Рассела и Уайтхеда, Тарского, исчисление предикатов и другие ранние фундаментальные формализмы при необходимости могут быть реализованы в правилах (отношениях) миварных сетей. Надо только помнить о том, что существуют и другие формализмы представления информации для ИИ, а у исчисления предикатов первого порядка есть достаточно жесткие ограничения и не очень большие возможности, если говорить о создании глобальных познающе-диагностических систем и СПАКОД (самоорганизующихся программно-аппаратных комплексов оперативной диагностики), решающих сложные логико-вычислительные задачи в реальном времени.

3. ВОЗМОЖНОСТИ И ПРЕИМУЩЕСТВА МИВАРНОГО ПОДХОДА

3.1. Возможности миварного подхода

Разум человека одновременно накапливает и хранит данные, комплексно решает и логические, и вычислительные, и логико-вычислительные задачи. Миварный подход позволяет в едином формализме проводить эволюционное накопление данных в миварном информационном пространстве, а также выполнять совмещенную логико-вычислительную обработку в миварных логических сетях. Поэтому использование миварной технологии логико-вычислительной обработки дает возможность решать большой класс сложных научных и практических задач. В главе 1 был проведен анализ существовавших ранее подходов к решению различных классов задач.

В этой главе проводится анализ проблем, достижений и перспектив в области баз данных и миварном информационном пространстве унифицированного представления данных и правил.

Проблема создания интеллектуальных систем остается актуальной и практически значимой [1–6, 8–22]. Миварные сети открывают принципиально новые возможности в сфере разработки действительно интеллектуальных и логически рассуждающих систем (ЛРС, Разуматоров), а также позволяют перейти к новому поколению экспертных систем, автоматизированных систем управления (АСУ), систем поддержки принятия решений (СППР), распознавания образов, понимания, аннотирования и перевода текста, поисковых систем, при разработке роботов и др. Путем объединения баз данных с логико-вычислительной обработкой миварный подход позволяет работать с огромными объемами контекстной информации, что порождает способность

компьютеров к непосредственному пониманию смысла информации и дает предпосылки создания логического искусственного интеллекта (ЛИИ). Миварный путь создания ИИ можно показать так:

Миварные технологии \Rightarrow *БД + Логика* \Rightarrow *Контекст* \Rightarrow
 \Rightarrow *Смысл* \Rightarrow *Логический искусственный интеллект*.

Таким образом, миварные сети обуславливают переход к новому поколению экспертных систем и интеллектуальных пакетов прикладных программ. Миварный подход позволил на практике создать автоматические, обучаемые, эволюционные, активные, логически рассуждающие информационные системы. В перспективе на основе миварных сетей будет создана глобальная мультипредметная активная экспертная система под названием «Миварная активная энциклопедия».

3.2. Преимущества миварного подхода перед семантическими сетями и продукциями

С точки зрения обоснования преимуществ и перспективности миварного подхода важно сделать следующее замечание. Продвижение в решении задачи создания искусственного интеллекта напрямую зависит от выбора способа представления информации. Выбор подходящего представления весьма важен для разработчиков компьютерных программ, обеспечивающих решение задач, относящихся к сфере искусственного интеллекта. Несмотря на большое разнообразие языков представления, которые можно использовать для создания искусственного интеллекта, подходящий язык должен отвечать общим требованиям выразительности, эффективности и правильности дедуктивных выводов. Выбор и оценка языков представлений — весьма важная задача для исследователей и программистов.

Выразительность миварного подхода ни в чем не уступает ни исчислениям предикатов, ни семантическим сетям, ни другим известным формализмам в области ИИ. Более того, изменяю-

3. Возможности и преимущества миварного подхода

щееся многомерное миварное информационное пространство позволяет в едином формализме описать и совместить все указанные формализмы. В частности, миварное пространство дает возможность отобразить семантическую сеть в многомерном пространстве, что только увеличивает ее выразительность и позволяет добавить новые связи за счет многомерности. С онтологиями можно провести аналогичную семантическим сетям процедуру. Даже модель данных «сущность–связь» можно легко представить в миварном пространстве [1, 4].

Исчисление предикатов имеет равные выразительные способности с семантическими сетями. Следовательно, по выразительности миварный подход может объединять в себе все перечисленные формализмы, что значительно увеличивает его выразительность.

В семантических сетях и исчислениях предикатов процесс обработки информации носит явно выраженный NP-полный характер. Это обусловлено тем, что вся обработка ведется на основе теории графов путем применения «графа пространства состояний». Но несмотря на эту универсальность, поиска в пространстве состояний не достаточно для автоматизации интеллектуального поведения, обеспечивающего автоматическое решение проблем. Если бы поиска в пространстве состояний было достаточно, то нужно было бы осуществлять полный поиск по всему пространству состояний. Этот метод известен как «исчерпывающий поиск» или «поиск методом полного перебора». Хотя полный перебор может применяться в любом пространстве состояний, огромный размер пространства для сложных задач делает этот подход практически неприемлемым. Поиск обеспечивает структуру для автоматизации решения задач, но эта структура лишена интеллекта. Такой подход не дает возможности формально описать задачу. Кроме того, простой метод полного перебора для больших предметных областей вообще практически неосуществим и непригоден для описания сущности разумной деятельности.

Получается, что *графовое представление семантических сетей не всегда позволяет эффективно и разумно осуществлять обработку информации и поиск решений в сложных задачах*. Для решения вышеописанной проблемы и создавался миварный подход как альтернатива графовому подходу с полным перебором (NP-полнота).

Миварный подход позволяет находить решение с линейной вычислительной сложностью для очень многих задач.

Этот подход базируется на представлении информации в виде специальной миварной матрицы и отличается от традиционного логического вывода тем, что не используется исчисление предикатов и не рассматриваются в явном виде вопросы истинности или ложности высказываний. Для реальных задач достаточно построить алгоритм решения из модулей (сервисов, отношений и т.п.), где на основе входных данных можно вычислить или определить требуемые выходные данные.

Такой подход возможен, поскольку с точки зрения познающе-диагностических систем вопросы ложности высказываний в исчислениях предикатов не имеют практического смысла. Для выполнения задач познания и диагностирования требуется только алгоритм вывода от входа к выходу, который на миварной матрице ищется за один проход и без циклов.

На данном этапе не выявлено ограничений описаний предметных областей для реальных задач с позиции возможности применения миварных матриц в случае, если задачи возможно сформулировать в терминах продукций при выделении условий и действий в основном формате продукций «если..., то...». А продукции, как известно, в общем виде позволяют представить любые причинно-следственные (каузальные) зависимости. Продукционные системы, основанные на представлении знаний в виде продукционных правил, являются обобщенной и мощной моделью решения задач на основе поиска.

3. Возможности и преимущества миварного подхода

Одной из стратегий решения сложных системных задач является описанная Дж. Люгером методология «классной доски» — решения сложных задач с привлечением разнородных источников знаний, взаимодействующих через общее информационное поле [10]. Именно эта методология используется и в миварном подходе для решения самого разного класса задач [1].

В экспертных системах, в которых используются продукционные правила, модификация одного правила не имеет глобальных синтаксических побочных эффектов. Правила могут добавляться или удаляться без необходимости изменения всей программы, а легкость модификации базы знаний является главным фактором производства успешных программ.

В экспертной системе, *основанной на правилах*, знание представляется в форме правил «*если..., то...*». База знаний содержит как общие знания, так и информацию о частных случаях. Механизм вывода применяет знания при решении реальных задач. По существу он является интерпретатором базы знаний. В продукционной системе механизм вывода совершает цикл «распознавание–действие». Процедуры, которые выполняет этот управляющий цикл, отделены от самих продукционных правил. Кроме того, в системе, основанной на правилах, пары «условие–действие» представляются правилами «*если..., то...*», в которых посылка (часть «если») соответствует условию, а заключение (часть «то») — действию. Если условие удовлетворяется, экспертная система осуществляет действие, предусмотренное при истинности заключения. Данные частных случаев можно хранить в рабочей памяти. При этом управление может осуществляться либо на основе данных, либо на основе цели.

Таким образом, если говорить о продукционных системах и системах на основе миварных сетей, можно употреблять термин «логический вывод», но в контексте цикла «распознавание–действие».

Очень важно, что экспертные системы строятся методом последовательных приближений. Выявляемые ошибки приводят к коррекции и наращиванию базы знаний. Программа никогда не должна рассматриваться как законченная. Эвристическая база знаний всегда будет иметь ограниченные возможности. Модульность модели продукционной системы позволяет естественным образом добавить новые правила или в любое время подкорректировать существующую базу правил. Все вышесказанное относится и к миварным сетям. Важные различия между миварными сетями и продукциями начинаются на этапе обработки информации.

Главным ограничением применения продукций является то, что продукционная система осуществляет полно-переборный **поиск на графе**. Программы подсистемы объяснений отслеживают процесс поиска на графе и используют эту информацию, чтобы отвечать на вопросы пользователя. С помощью продукционных правил каждый шаг процесса рассуждений документируется автоматически.

Миварный подход развивает продукционный в самом общем смысле, он позволяет включать в виде правил любые отношения из миварного информационного пространства, которое в свою очередь обобщает модель семантических сетей и модель «сущность–связь».

Более того, применяя современные технологии многоагентных систем и сервисно-ориентированных архитектур, миварный подход дает возможность включать в виде отношений и сервисы, и традиционные вычислительные процедуры, и процедуры ГРИД-систем, и подобные отношения, функции и правила.

Философские основания миварного подхода гарантируют самые широкие возможности по описанию и представлению в миварных сетях практически любых видов отношений в самом широком смысле.

3. Возможности и преимущества миварного подхода

Для экспертных систем на основе продукционного подхода и вместо традиционных однодольных графов необходимо и достаточно использовать двудольные графы миварных сетей. С научно-практической точки зрения основную сложность при использовании экспертных систем составляет именно концептуальное продукционное описание предметной области и формирование необходимых двух списков: объектов и правил для миварных сетей. Непосредственная обработка проводится на основе универсального механизма.

Выделенные правила-процедуры при необходимости могут быть представлены в виде продукций, что соответствует традиционному подходу, но в то же время позволяют реализовать многомерное эволюционное представление данных. Неоспоримыми преимуществами миварного подхода являются:

- линейная вычислительная сложность и реальное время работы;
- решение логических и вычислительных (и других) задач;
- управление потоком входных данных и оперативная диагностика;
- адаптивное описание и непрерывное решение задач;
- активная работа с запросами или уточнениями входных данных на эволюционной сети правил и объектов (самообучение).

4. МИВАРНЫЙ МЕТОД ЛОГИКО-ВЫЧИСЛИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ

4.1. Три этапа миварной обработки информации

Для реализации метода логико-вычислительной обработки данных необходимо сделать следующие шаги. Прежде всего должно быть разработано формализованное описание предметной области. Для этого на основе миварной технологии представления информации выделяют объекты-переменные и правила-процедуры, а затем создают соответствующие списки объектов и правил (табл. 2.1, рис. 2.1). Формализованное представление метода логико-вычислительной обработки данных аналогично двудольному графу миварной логической сети, показанному на рис. 2.2.

Таблица 4.1. Пример исходного задания матрицы А размерности $(m + 1) \times (n + 1)$ со структурой сети правил

А	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	
2			X	Y	Y			X	X	
...										
m		X		X	X		Y			
m+1										

Выделим три основных этапа миварной обработки информации:

- 1) формирование миварной матрицы А описания предметной области;
- 2) работа с матрицей и конструирование алгоритма решения заданной задачи;

4. Миварный метод логико-вычислительной обработки данных

3) выполнение всех вычислений по полученному алгоритму и нахождение ответа.

Первый этап по существу является формализацией предметной области (табл. 4.1 и рис. 4.1) (обозначения параметров X, Y, Z, W в табл. 4.2–4.8 и рис. 4.2–4.8 приведены в табл. 2.2).

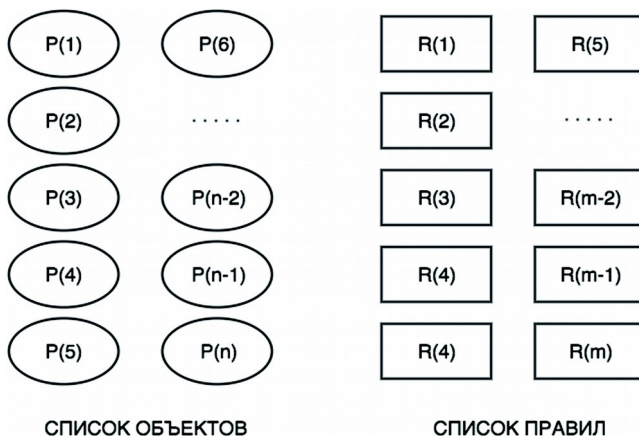


Рисунок 4.1. Пример исходного задания двудольного графа миварной логической сети

Этот этап в настоящее время является наиболее сложным и требует участия человека-специалиста (эксперта) для создания миварной модели предметной области.

На втором этапе выполняется автоматическое конструирование алгоритма решения или логический вывод (табл. 4.2–4.8, рис. 4.2–4.8). Исходными данными для этого являются миварная матрица описания предметной области, а также заданные входные («дано») и искомые («найти») объекты-переменные (табл. 4.2 и рис. 4.2).

На третьем этапе осуществляется решение по полученному алгоритму. Ниже будет проиллюстрирована работа алгоритма.

Таблица 4.2. Пример обработки матрицы А. Шаг № 1. Постановка задачи

A	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	
2			X	Y	Y			X	X	
...						...				
m		X		X	X		Y			
m+1		Z	Z	Z			W			

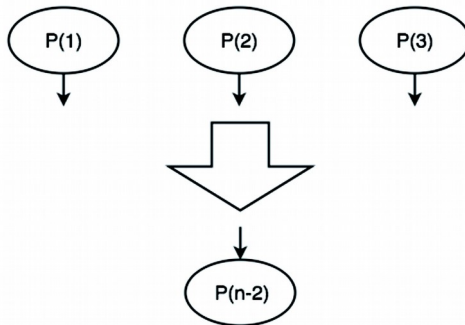
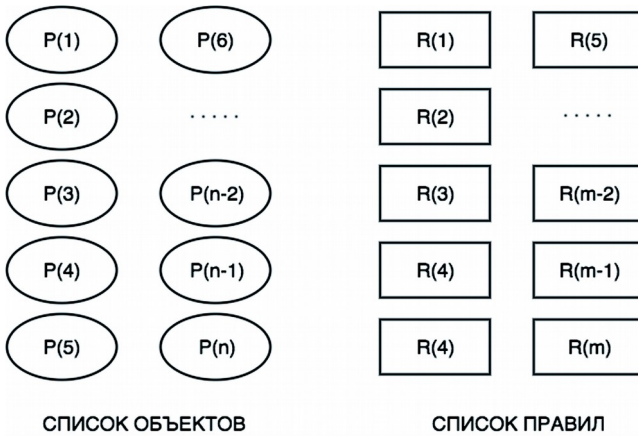
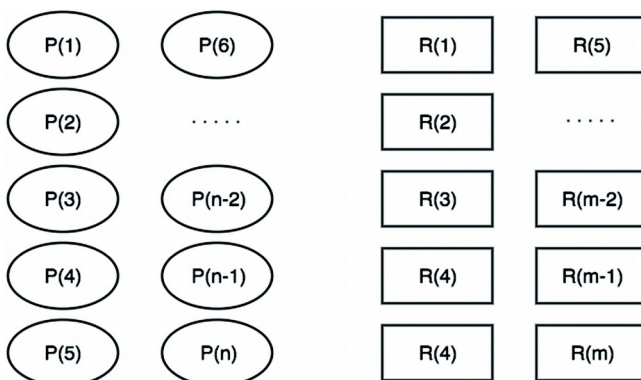


Рисунок 4.2. Шаг № 1. Двудольный граф

4. Миварный метод логико-вычислительной обработки данных

Таблица 4.3. Пример обработки матрицы А. Шаг № 2

A	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	1
2			X	Y	Y			X	X	
...						...				
m		X		X	X		Y			
m+1	Z	Z	Z				W			



СПИСОК ОБЪЕКТОВ

СПИСОК ПРАВИЛ

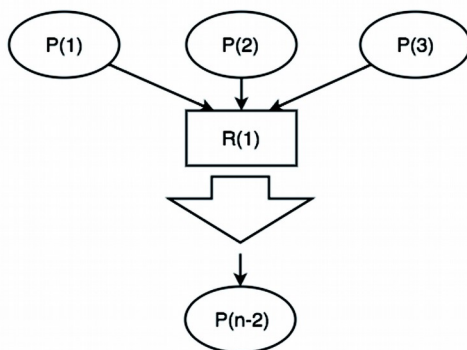
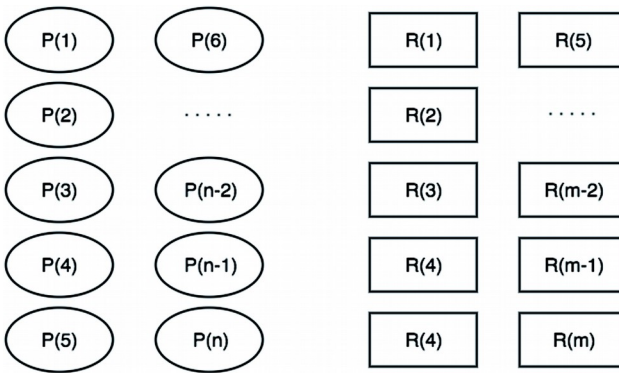


Рисунок 4.3. Шаг № 2. Двудольный граф

Таблица 4.4. Пример обработки матрицы А. Шаг № 3

A	1	2	3	4	5	...	n-2	n-1	n	n+1	
1	X	X	X						Y	Y	2
2			X	Y	Y			X	X		
...						...					
m		X		X	X		Y				
m+1	Z	Z	Z				W	Z	Z		



СПИСОК ОБЪЕКТОВ

СПИСОК ПРАВИЛ

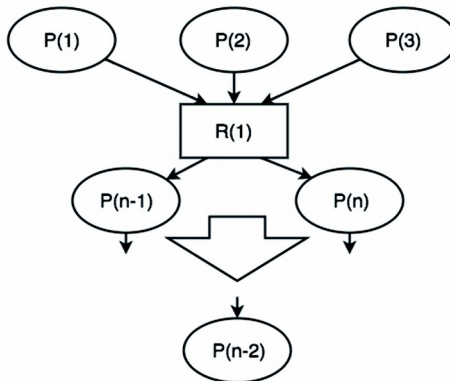
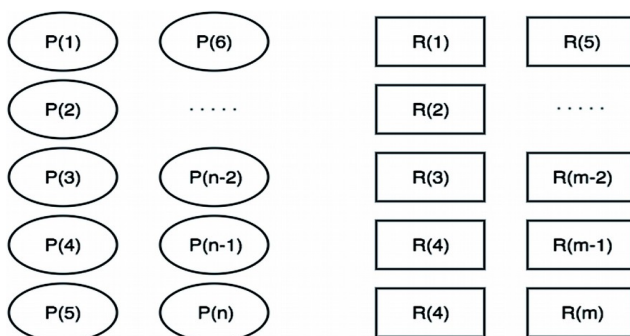


Рисунок 4.4. Шаг № 3. Двудольный граф

4. Миварный метод логико-вычислительной обработки данных

Таблица 4.5. Пример обработки матрицы А. Шаг № 4

A	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	2
2			X	Y	Y			X	X	1
...						...				
m		X		X	X		Y			
m+1	Z	Z	Z				W	Z	Z	



СПИСОК ОБЪЕКТОВ

СПИСОК ПРАВИЛ

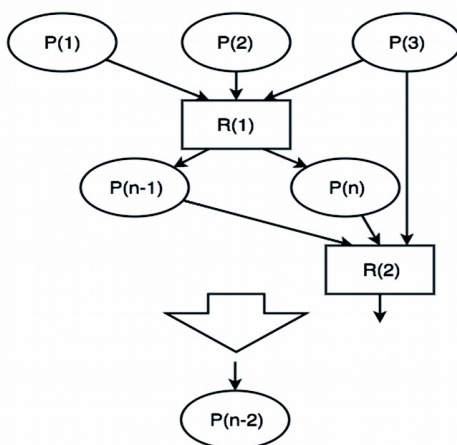
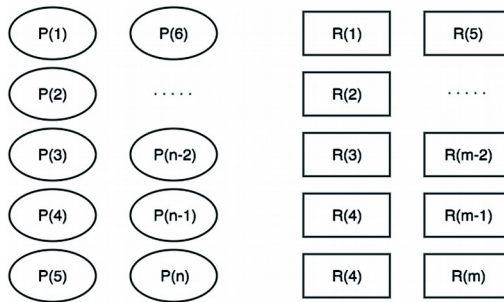


Рисунок 4.5. Шаг № 4. Двудольный граф

Таблица 4.6. Пример обработки матрицы А. Шаг № 5

A	1	2	3	4	5	...	n-2	n-1	n	n+1	
1	X	X	X						Y	Y	2
2			X	Y	Y			X	X	2	
...						...					
m		X		X	X		Y				
m+1	Z	Z	Z	Z	Z		W	Z	Z		



СПИСОК ОБЪЕКТОВ

СПИСОК ПРАВИЛ

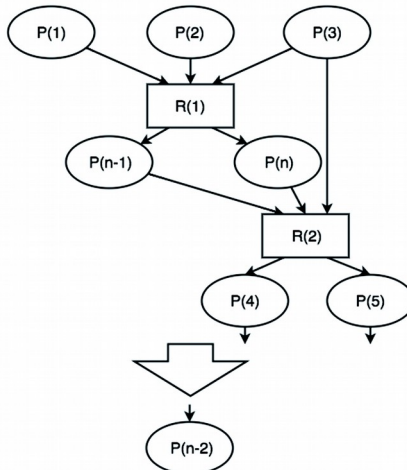
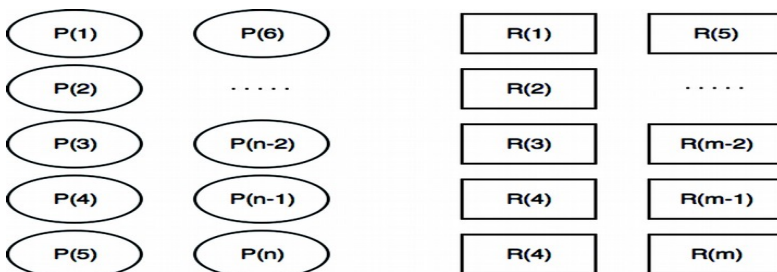


Рисунок 4.6. Шаг № 6. Двудольный граф

4. Миварный метод логико-вычислительной обработки данных

Таблица 4.7. Шаг № 6 примера обработки матрицы V

V	1	2	3	4	5	...	n-2	n-1	n	n+1	
1	X	X	X						Y	Y	2
2			X	Y	Y				X	X	2
...						...					
m		X		X	X		Y				1
m+1	Z	Z	Z	Z	Z		W	Z	Z		



СПИСОК ОБЪЕКТОВ

СПИСОК ПРАВИЛ

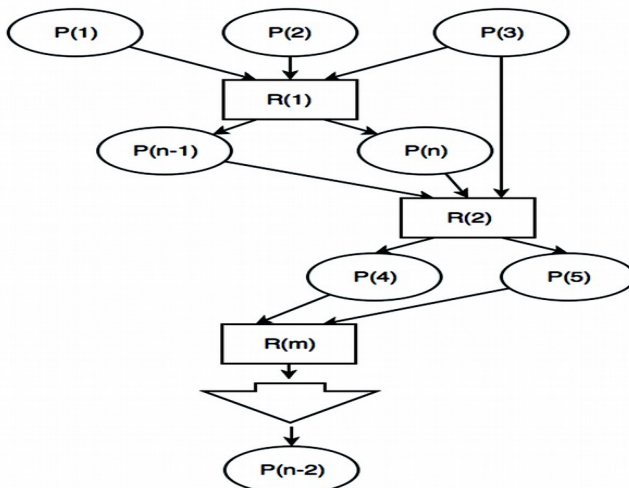


Рисунок 4.7. Шаг № 6. Двудольный граф

Таблица 4.8. Пример обработки матрицы А. Шаг № 7

A	1	2	3	4	5	...	n-2	n-1	n	n+1
1	X	X	X					Y	Y	2
2			X	Y	Y			X	X	2
...						...				
m		X		X	X		Y			2
m+1	Z	Z	Z	Z	Z		Z (W)	Z	Z	

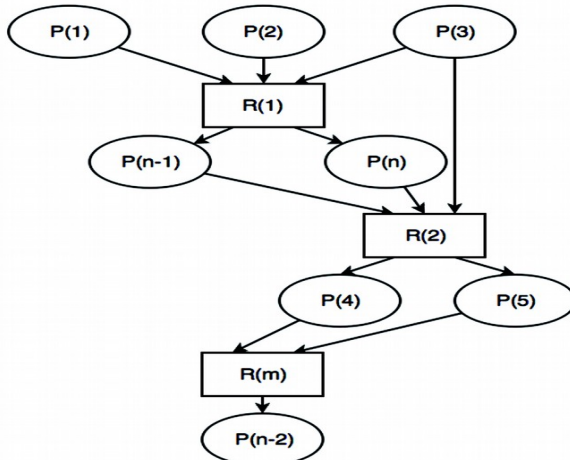
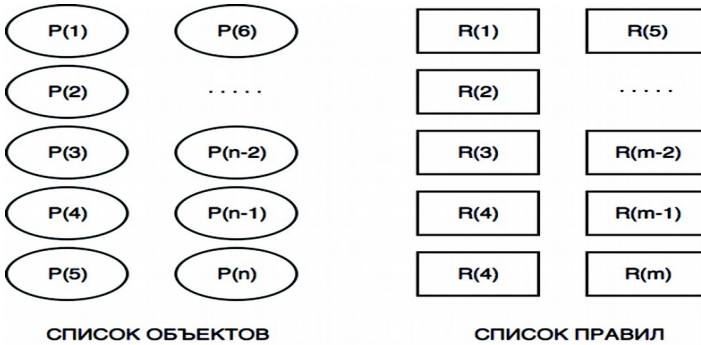


Рисунок 4.8. Шаг № 7. Двудольный граф

4. Миварный метод логико-вычислительной обработки данных

Прежде всего, как уже было отмечено, для миварной сети логических правил, представленной в списочной форме, строится матрица. Затем на основе анализа этой матрицы определяется факт наличия успешного маршрута вывода, потом определяются возможные маршруты логического вывода, а на последнем этапе из этих маршрутов выбирают кратчайший, наиболее оптимальный по заданным критериям.

Пусть известны m правил и n переменных (входящих в правила либо в качестве исходных, активизирующих их, либо в качестве получаемых, т.е. выходных). Тогда в матрице $A(m \times n)$, каждая строка которой соответствует одному из правил и содержит информацию об используемых в правиле переменных, могут быть представлены все взаимосвязи между правилами и переменными. При этом в каждой строке все входные переменные этого правила на соответствующих позициях матрицы помечаются символом X , все выходные — Y ; все переменные, которые уже получили в процессе вывода или задания исходных данных конкретное значение, — Z ; а все искомые (выходные) переменные, то есть те, которые необходимо вывести из исходных (входных) данных, — W . Кроме того, в матрицу A добавлена одна строка и один столбец для хранения в них служебной информации. Стоит напомнить, что количество служебной информации может видоизменяться для различных конкретных реализаций программных комплексов. Получаем матрицу A размерности $(m + 1) \times (n + 1)$, в которой отражена вся структура исходной сети правил. Стоит отметить, что структура этой логической сети может изменяться в любое время, т.е. это сеть правил с эволюционной структурой. Пример такой матрицы показан в табл. 4.1. На рис. 4.1 эта же матрица продемонстрирована в виде двудольного графа миварной логической сети.

Для поиска маршрута логического вывода на полученной матрице производят действия согласно следующим этапам.

1. В строке $(m + 1)$ известные переменные помечают Z , а искомые — W . Например, в табл. 4.2 в строке $(m + 1)$ символом Z отмечены позиции 1, 2, 3, а символом W — позиция $(n - 2)$. На рис. 4.2 это действие показано с помощью двудольного графа миварной сети.
2. Далее осуществляется последовательно, например сверху вниз, поиск таких правил, которые могут быть активизированы, т.е. таких, у которых известны все входные переменные. Если таких правил нет, то маршрута логического вывода нет и необходимо выдать запрос на уточнение (добавление) входных данных. Если такие правила, которые могут быть активизированы, есть, то у каждого из них в соответствующем месте служебной строки делается пометка, что правило может быть запущено. Например, можно ставить в этих клетках таблицы цифру 1, что и показано в табл. 4.3 в клетке $(1, n + 1)$. На рис. 4.3 это действие показано на примере двудольного графа миварной логической сети.
3. Если таких правил несколько, то осуществляется выбор по заранее определенным критериям такого или таких правил, которые должны быть активизированы в первую очередь. При наличии достаточных ресурсов одновременно могут запускаться сразу несколько правил.
4. Далее производится имитация запуска правила (процедуры), путем присваивания выводимым в этом правиле переменным значений «известно», в нашем случае, — Z . Запущенное правило для удобства дальнейшей работы помечается дополнительно цифрой 2 (это не обязательно). Соответствующие изменения показаны в табл. 4.4 (см. элементы $(m + 1, n - 1)$, $(m + 1, n)$ и $(1, n + 1)$). На рис. 4.4 это действие показано на примере двудольного графа миварной логической сети.
5. После имитации запуска правил проводят анализ достижения цели, т.е. анализируют получение требуемых значений путем сравнения служебных символов в служебной строке. Если в служебной строке $(m + 1)$ осталось хоть одно значение

4. Миварный метод логико-вычислительной обработки данных

«искомая» (W), то осуществляют дальнейший поиск маршрута логического вывода. В противном случае задача считается успешно решенной, а все задействованные правила в соответствующем порядке их запуска и образуют искомый маршрут логического вывода. В табл. 4.4 и на рис. 4.4 видно, что искомые значения не выявлены, поэтому необходимо переходить к следующему этапу.

6. Прежде всего определяют наличие таких правил, которые могут быть запущены после определения новых значений на предыдущем этапе. Если таких правил нет, то маршрута вывода нет и поступают аналогично пункту 2 этого метода. Если такие правила есть, то продолжают поиск маршрута вывода. В приведенном примере такие правила есть, что показано в табл. 4.5. В клетке $(2, n + 1)$ появилась 1 как признак возможности запуска данного правила. На рис. 4.5 это действие показано на примере двудольного графа миварной логической сети.

7. На следующем этапе аналогично этапу 4 запускают правила (имитируют запуск), далее аналогично этапам 5 и 6 выполняют необходимые действия столько раз, сколько требуется для получения результата. При необходимости повторяют этапы 2–7 до достижения результата. При этом он может быть как положительный — маршрут вывода существует, так и отрицательный — вывода нет из-за неопределенности входных данных. Для наглядности будет продолжено пошаговое выполнение рассматриваемого примера. Итак, необходимо провести имитацию запуска правила 2, что и показано в табл. 4.6. На рис. 4.6 это действие показано на примере двудольного графа миварной логической сети.

8. В клетках $(m + 1, 4)$ и $(m + 1, 5)$ получаем признак выводимости переменных 4 и 5, а в клетке $(2, n + 1)$ формируется признак того, что правило уже было запущено, т.е. ставится цифра 2. После этого нужно провести анализ служебной строки, тогда можно увидеть, что не все искомые переменные известны.

Значит, необходимо продолжить обработку матрицы A размерности $(m + 1) (n + 1)$. Анализ этой матрицы показывает возможность запуска правила m , что и отражено в табл. 4.7 и на рис. 4.7.

9. При дальнейшей обработке примера при запуске правила m появляются новые значения, в том числе и для искомым переменных, что и показано в табл. 4.8 и на рис. 4.8.

10. В результате получено, что в примере в служебной строке больше не осталось искомым переменных, а в клетках таблицы появились новые значения: в клетке $(m, n + 1)$ — 2, а в клетке $(m + 1, n - 2)$ вместо значения W появилось значение Z . Таким образом, получен положительный результат, следовательно, маршрут логического вывода при данных исходных значениях существует.

4.2. Теоретический расчет вычислительной сложности миварного метода обработки

Общее количество действий при использовании миварного метода определяется суммой действий на каждом этапе:

- 1) присваивание известных Z и искомым W значений клеткам служебной строки $(m + 1)$ (общее количество таких действий не более n);
- 2) присвоение признака обработки правил в служебном столбце $(n + 1)$ (количество действий не более $2m$, но может быть не более числа m);
- 3) присваивание признака известности Z клеткам служебной строки $(m + 1)$ (общее количество таких действий не более n);
- 4) определение новых значений клеткам строки $(m + 1)$ (количество таких действий не более n).

4. Миварный метод логико-вычислительной обработки данных

Действия на этапах 1, 2 и 3 выполняются над одним массивом данных — строкой ($m + 1$). При этом общее суммарное количество действий на всех этих этапах не должно превышать общего количества клеток в этой строке, так как обработанные значения «вычеркиваются» и более не обрабатываются. Получается, что при миварном методе обработки общее количество действий (KD) (его вычислительная сложность), не превышает количества клеток в служебных частях матрицы: $O(m + n)$, т.е. $KD = O(m + n)$.

В том случае, когда нельзя реализовать предложенные сокращения вычислений, миварный метод решает задачу поиска маршрута логического вывода с вычислительной сложностью:

$$KD = O(mn).$$

То есть в худшем случае по каждой клетке миварной матрицы проходят один раз.

Таким образом, относительно классического подхода получаем линейную зависимость вычислительной сложности от количества правил (где $n = \text{Const}$):

$$KD = O(m \times \text{Const}).$$

Если $m = \text{Const}$, то $KD = O(\text{Const} \times n)$. При решении практических задач часто количество правил и переменных не изменяется, тогда получается, что:

$$KD = O(\text{Const} \times \text{Const}).$$

4.3. Научно-популярная иллюстрация применения миварного подхода

На рис. 4.9–4.10 популярно показана работа с миварной сетью:

- 1) создание миварной сети;
- 2) успешное нахождение маршрута логического вывода.

На первой серии рисунков «Создание миварной сети» (рис. 4.9) отражено, как группа пользователей параллельно выявляет миварные правила и заносит их в базу данных, постепенно создавая из них полноценную и довольно объемную миварную сеть. Важно, что эта работа происходит параллельно и достаточно быстро. Желтые узлы символизируют объекты-сущности, а линии — это правила их взаимодействия или связи между объектами. В отличие от миварной сети обычные описания для семантических сетей и онтологий, как правило, надо делать последовательно и только одному «инженеру по знаниям» путем формирования концептуальной модели предметной области.

На второй серии рисунков «Успешный поиск маршрута вывода на миварной сети» (рис. 4.10) показано, как двое пользователей начинают поиск маршрута логического вывода: первый тянет сеть за входные данные, а второй в другую сторону, за выходные (целевые показатели, объекты). Вследствие этого в процессе вывода задействуются только те объекты и правила, которые необходимы для данного поиска маршрута логического вывода. Таким образом удается избежать полного перебора с циклами, свойственного традиционным механизмам вывода, основанным на предикатах или графах. В случае существования маршрута вывода он быстро находится — пользователи его «вытягивают», и встречаются друг с другом, что отмечено смайликом.

Когда нет маршрута логического вывода, а входные объекты и выходные объекты находятся в разных и невзаимосвязанных сегментах сети, пользователи начинают «вытягивать» маршрут вывода, но обнаруживается, что концы сети не связаны между собой, т.е. маршрута вывода нет. Если говорить математическим языком, это ситуация, когда минимальный разрез многоплюсной сети между входом и целью получается равным нулю (нет связи).

4. Миварный метод логико-вычислительной обработки данных

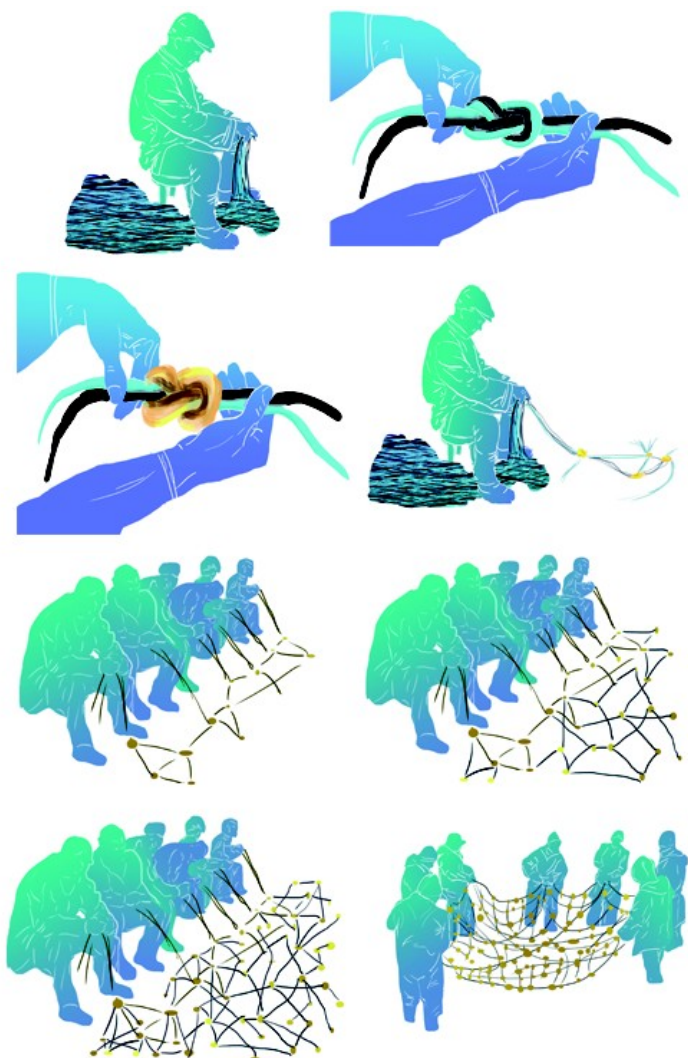


Рисунок 4.9. Создание миварной сети

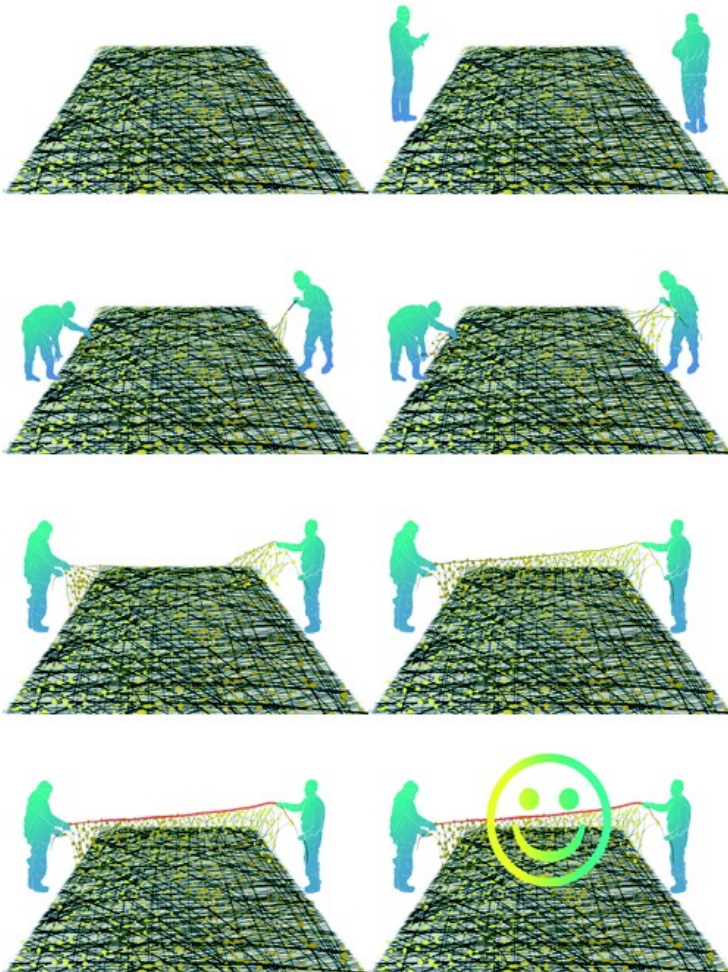


Рисунок 4.10. Успешный поиск маршрута вывода на миварной сети

Даже в таком варианте у миварных сетей есть важные преимущества: видно, где произошел разрыв и автоматически определя-

ется, каких именно входных объектов не хватает для продолжения поиска маршрута логического вывода.

Следовательно, в познающе-диагностических системах можно подать сигналы на систему датчиков или каким-либо другим образом запросить необходимые для дальнейшей работы входные данные.

Важно отметить, что в миварной сети представляется возможность создания механизма (алгоритма) поиска логического вывода с линейной вычислительной сложностью, а накопление огромного количества информации реализуется в современных базах данных (для которых это не является большой проблемой). Именно за счет хранения всей информации по каждому элементу миварной сети возникает возможность прямого и однозначного поиска логического вывода с линейной вычислительной сложностью и исключение переборных и возвратных, что было характерно для предшествующих схем логического вывода с NP-полной сложностью поиска решения.

В настоящее время Конструктор экспертных систем миварный Wi!Mi РАЗУМАТОР 2.1. выполняет поиск логического вывода и автоматически конструирует управляемые потоком входных данных алгоритмы решения задач из готовых модулей-сервисов. На обычном ноутбуке Конструктор экспертных систем миварный Wi!Mi РАЗУМАТОР 2.1. обрабатывает более 5 млн продукционных правил менее чем за 0,05 секунды

Программная реализация миварных сетей в виде КЭСМИ наглядно доказывает на практике линейную сложность поиска логического вывода, эволюционность и активность работы миварных экспертных систем нового поколения.

4.4. Сказка: паровоз, искусственный интеллект и Старик Хоттабыч

Попытки создания систем искусственного интеллекта напоминают известную детскую сказку о волшебнике старике Хоттабыче. Представим себе, что старик Хоттабыч увидел обыкно-

венный паровоз, который тянет за собой огромные тяжелогруженные вагоны. Он видит форму, размеры паровоза, но не знает ни его устройства, ни принципа работы, ни того, что человек-машинист управляет сложнейшей машиной. И вот захотел Хоттабыч сделать, разработать, создать такой паровоз. Ну, а что в этом такого сложного? Вот же перед нами паровоз, все видно, понятно и просто: дым из трубы идет, значит едет паровоз, а когда дыма нет, то паровоз стоит на месте.

Дернул старик Хоттабыч волос из своей бороды, «дзинь», и стоит новенький блестящий паровоз, из чистого золота сделанный, дорогами коврами увешанный. Из трубы паровоза дым валит, но паровоз не движется, стоит на одном месте и сверкает в лучах заходящего солнца. Хорошо, если Хоттабыч хотя бы знаком с устройством колес и смог воспроизвести их. Тогда этот паровоз можно будет по крайней мере хотя бы подталкивать и катать. А если удастся поднять, затащить его на горку, то тогда он даже сам покатится. Но не далеко, а только пока спуск не кончится. Потом снова можно поднять паровоз на горку, и снова, и снова...

Так же и человек: увидел интеллект, вроде бы выяснил, как он выглядит, и решил сделать искусственный интеллект. Собрал, сделал. И вроде бы внешне все есть у этого интеллекта. Разрабатывая различные алгоритмы и создавая программы, человек поднимает свое творение — искусственный интеллект на «горку». Чем больше и чем лучше алгоритмы, тем выше такая «горка», тем дальше может самостоятельно скатываться, т.е. решать задачи, паровоз-интеллект, причем без помощи человека. Рукоплещут зрители, кричат мальчишки, взволнованно дышат барышни: «Смотрите, смотрите! Машина сама, сама решает задачи!! Браво!!!» Победа? Теперь еще можно добавить, что компьютер обыгрывает человека в шахматы и т.п. Но кончилась «горка», и оказывается, что не машина сама, а человек с помощью машины решал те самые задачи, разрабатывал алгоритмы, писал программы. Какой конфуз! Какая неудача!! Какой провал!!!

Сейчас, наука пошла дальше в изучении паровоза-интеллекта. Известно уже, зачем ему колеса, почему дым идет из трубы. Вроде бы теоретически понятно, как работает двигатель. Обнаружено, что любая строго ограниченная область мыслительной деятельности человека в принципе может быть формализована, описана алгоритмически и, следовательно, воспроизведена, повторена, проделана машиной или ЭВМ. Лихо и шустро, с бравым видом, с блеском скатываются «паровозы» с высоких, крутых алгоритмических и программных «гор». Естественно, скатываются они по рельсам, которые проложил человек. Но, задумался уже человек: «Кто же управляет паровозом?». Вот стоит человек и рассуждает: «Кто (или что) управляет мышлением? Как оно возникло?». А ответ пока только такой: раз сам сделал паровоз-интеллект, то сам и управляй им, пока ничего лучше не придумано.

Вот так и стал человек «машинистом» искусственного интеллекта. Сам дорогу выбирает, сам уголь в топку подбрасывает. Паровоз же пыхтит, дым в небо пускает и тянет, тянет тяжелый груз науки, который стал уже непосилен для неоснащенного, не вооруженного никакой техникой человека. Когда же перестанет удовлетворять паровоз — слишком медленно, да и груз все тяжелее и тяжелее, не справляется паровоз, тогда человек сядет, задумается и создаст нечто более мощное, скоростное — электровоз. Полезай человек опять в кабину и в путь по нелегкому пути познания. Не будет справляться электровоз? Тогда, человек, создай что-нибудь более пригодное. Вот такой трудный путь. Все в твоих руках, Человек!

5. РУКОВОДСТВО ПО КЭСМИ РАЗУМАТОР Wi!Mi 2.1

5.1. Описание программы

Конструктор экспертных систем миварный (КЭСМИ) РАЗУМАТОР Wi!Mi 2.1 (далее по тексту для краткости будет использоваться обозначение: «Wi!Mi») предназначен для создания специализированных информационных систем, способных в режиме реального времени решать сложные логические задачи, связанные с обработкой больших массивов информации. Wi!Mi позволяет вычислять искомые параметры из исходных путем нахождения алгоритмов вычислений. При этом производится расчет всех необходимых промежуточных параметров.

Wi!Mi в процессе вычисления рассчитывает только необходимые параметры, связанные в алгоритмические цепочки вычислений, а не все возможные параметры. Благодаря такому подходу достигается высокая производительность программы при небольших затрачиваемых ресурсах.

Технология *runtime*, добавленная во второй версии Wi!Mi, при построении алгоритма решения задачи учитывает значения рассчитанных промежуточных параметров.

Программа обеспечивает выполнение следующих функций:

- создание и редактирование качественных моделей ситуаций/предметных областей;
- структурный анализ моделей, получение логического вывода решения и объяснение его в виде последовательности выполненных действий.

5.2. Область применения

5.2.1. Краткое описание возможностей

Программа обеспечивает выполнение следующих функций:

- 1) создание и редактирование описания моделей предметных областей:

- объектов предметной области (параметров и классов);
 - отношений и правил, связывающих эти объекты;
- 2) структурный анализ, включающий оценку корректности и полноты введенных данных;
 - 3) анализ данных;
 - 4) создание и вывод полученного алгоритма логического вывода разрешения ситуации, расчет необходимых значений.

5.2.2. Уровень подготовки пользователя

Для работы с программой достаточно обладать знаниями о предметной области, для которой создается модель. Для создания сложного отношения (используются не часто) на языке JavaScript необходимо обладать навыками программирования.

5.3. Назначения и условия применения

5.3.1. Требования к аппаратным средствам

Программа может быть установлена на персональный компьютер с процессором 2GHz или более мощным, размером оперативной памяти не менее 4GB и свободным местом на жестком диске не менее 60MB.

5.3.2. Требования к программным средствам

Программа работает под управлением операционных систем Microsoft Windows 7, Microsoft Windows 8, Mac OS 10.9 и выше. Для работы со справкой по программе требуется Adobe Reader или иная программа, позволяющая читать файлы формата pdf.

5.4. Подготовка к работе

5.4.1. Состав и содержание программного пакета

В случае использования программного пакета на операционных системах Windows система моделирования КЭСМИ поставляется в виде файла WiMi_Demo.exe.

В случае использования программного пакета на операционных системах Mac OS система моделирования КЭСМИ поставляется в виде файла WiMi_Demo.dmg.

Эти файлы запускают процедуру инсталляции программы, руководства пользователя и методики описания предметных областей.

5.4.2. *Запуск системы*

Запуск программы осуществляется исполняемым файлом WiMi.app или WiMi.exe (в зависимости от операционной системы). Ярлык для указанного файла для удобства может быть создан пользователем на рабочем столе.

5.4.3. *Настройка*

Все модели, разрабатываемые в программе, хранятся в файле с расширением xml.

При загрузке программы открывается приложение, готовое к началу работы. Для загрузки существующей модели необходимо зайти в пункт «Файл» меню «Открыть», выбрать соответствующий файл xml и нажать кнопку «ОК». Одновременно в одной программе может быть открыта только одна модель.

5.4.4. *Этапы создания модели и моделирования*

Создание модели и моделирование в программе осуществляется в несколько этапов.

1. Создание и параметризация модели, подразумевающие выделение объектов (см. 5.5.2).
2. Создание и провязывание модели:
 - задание отношений (см. 5.5.3);
 - задание правил (см. 5.5.3.1);
 - задание условий корректности входных данных (ограничения) (см. 5.5.3.2).

3. Решение задач, текстовое описание выводов с указанием использованного правила, полученных и найденных данных.

5.5. Работа с программой

После загрузки модели все объекты будут располагаться в древо-видном списке в левой части приложения, а все отношения — в правой части. Редактирование соответствующих элементов модели производится во вкладках, расположенных в центральной части приложения. Если в процессе создания модели были обнаружены ошибки, соответствующие записи появятся в списке в нижней части приложения (рис. 5.1).

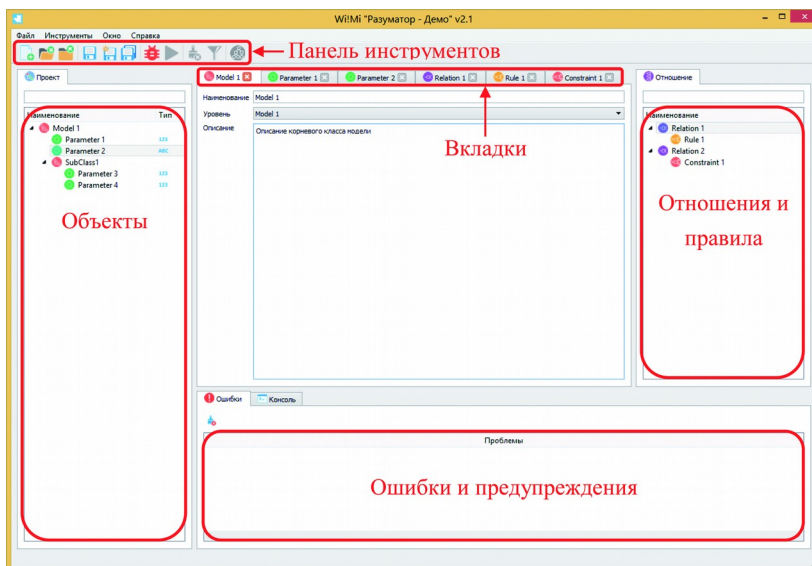


Рисунок 5.1. Основное окно приложения

5.5.1. Объекты

Существует два типа объектов: классы и параметры.

5.5.1.1. Классы

Класс – абстрактная сущность, обобщающее понятие. Класс может содержать в себе параметры и другие классы. Класс также имеет название, уровень иерархии и описание.

В любой модели должен существовать хотя бы один класс. Класс с высшим уровнем иерархии называют корневым. При создании новой модели, по умолчанию также создается корневой класс.

Чтобы создать, отредактировать или удалить класс, необходимо воспользоваться контекстным меню (рис. 5.2), вызываемым щелчком правой кнопкой мыши по существующему классу из древовидного списка в левой части окна.

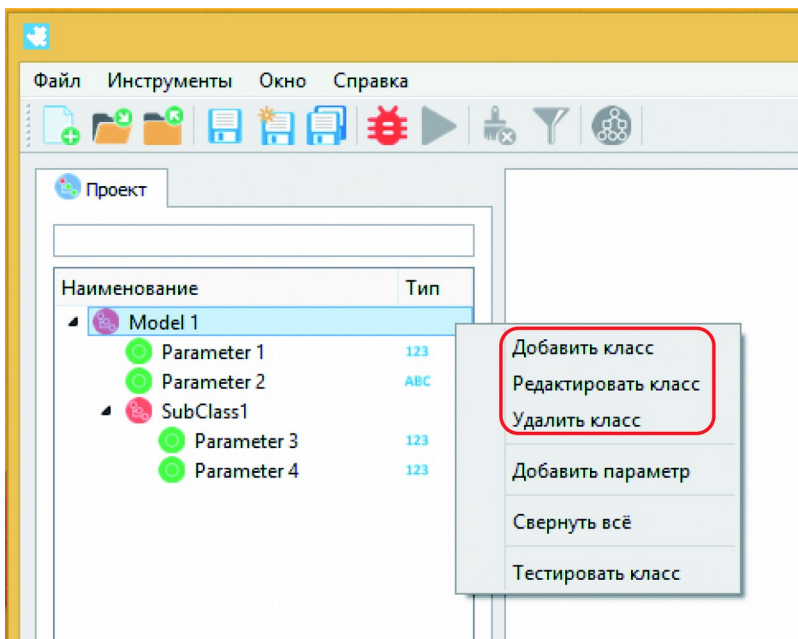


Рисунок 5.2. Контекстное меню, вызываемое по щелчку на классе

После нажатия кнопок «Добавить класс» или «Редактировать класс» откроется новая вкладка в центральной части окна, позволяющая совершить соответствующие операции.

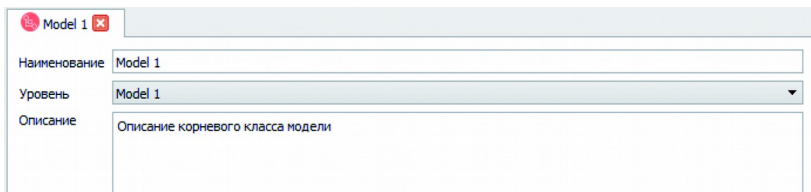


Рисунок 5.3. Вкладка, предназначенная для редактирования класса

В появившейся вкладке (рис. 5.3) пользователь может изменить название, поменять уровень иерархии, выбрав новый родительский класс из выпадающего списка, а также, по желанию, добавить описание.

Чтобы применить изменения или завершить процесс создания нового класса, сохраните модель. Новый или измененный класс будет отображаться в списке в левой части окна.

5.5.1.2. Параметры

Параметр — это объект, содержащий значение определенного типа: числового или текстового. Как и класс, параметр имеет название, уровень иерархии и описание. Кроме того, параметр может принимать какое-то значение по умолчанию.

Чтобы создать параметр, необходимо выбрать класс, внутри которого он будет создан, правым щелчком мыши по этому классу вызвать контекстное меню и выбрать пункт «Добавить параметр» (рис. 5.4). Для редактирования или удаления существующего параметра, необходимо воспользоваться контекстным меню, вызываемым щелчком правой кнопкой мыши по конкретному параметру из древовидного списка в левой части окна (рис. 5.5).

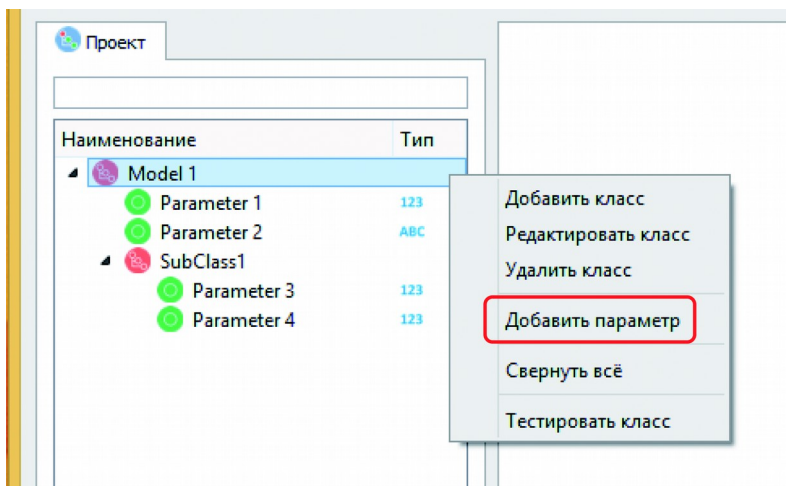


Рисунок 5.4. Контекстное меню для класса

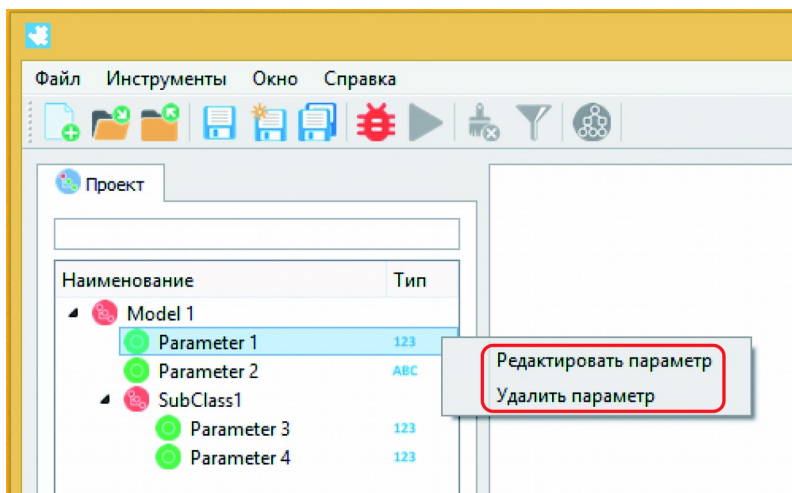


Рисунок 5.5. Контекстное меню для параметра

После нажатия кнопок «Добавить параметр» или «Редактировать параметр» откроется новая вкладка в центральной части

окна, позволяющая создавать новые параметры и вносить правки в уже существующие (рис. 5.6).

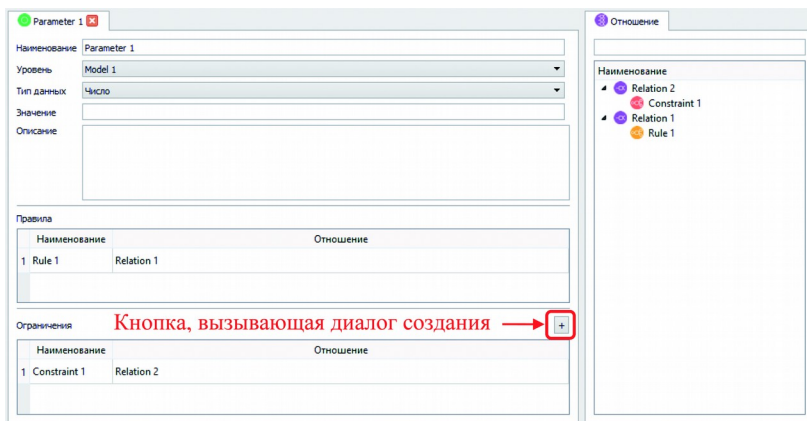


Рисунок 5.6. Вкладка, предназначенная для редактирования параметра

В появившейся вкладке пользователь может ввести наименование параметра, поменять уровень иерархии и тип данных. Помимо обязательных свойств пользователь также может добавить значение по умолчанию и описание. Кроме того, в данной вкладке существует две таблицы, отображающие связанные с конкретным параметром правила и ограничения. Каждая таблица состоит из двух столбцов: в первом отображается название, а в левом – отношение, с которым связано соответствующее правило или ограничение.

Чтобы применить изменения или завершить процесс создания нового параметра сохраните модель. Новый или измененный параметр будет отображаться в списке в левой части окна.

5.5.2. Отношения

При создании моделей предметных областей важную роль играют следующие понятия.

1. *Связь* — ассоциация между объектами, значимая для рассматриваемой предметной области. Произвольное количество входных объектов посредством связи преобразуется в один (или набор) выходных объектов.
2. *Отношение* — вид связи, использующий абстрактные переменные, описывающий их взаимодействие.
3. *Правило* — вид связи, привязывающий отношение к конкретным объектам.
4. *Ограничение* — вид правила, проверяющий входные данные на корректность. Например, сторона треугольника не может быть отрицательной.

Чтобы добавить, отредактировать или удалить отношение, вызовите контекстное меню, щелкнув на существующем отношении из списка в правой части окна приложения и выберите соответствующий пункт (рис. 5.7).

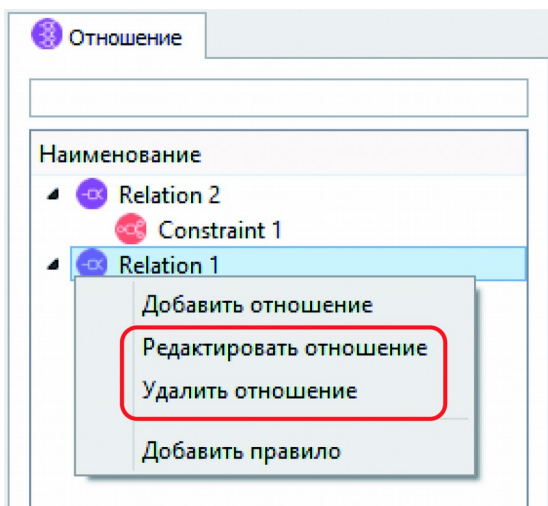


Рисунок 5.7. Контекстное меню для редактирования отношений

После нажатия кнопок «Добавить отношение» или «Редактировать отношение» откроется новая вкладка в центральной части окна, позволяющая совершить соответствующие операции (рис. 5.8).

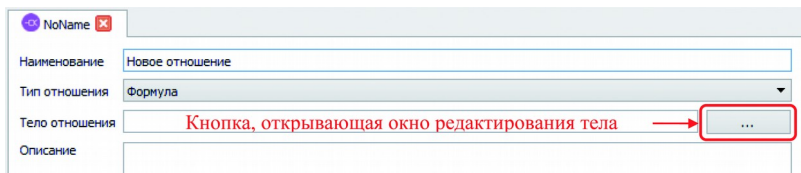


Рисунок 5.8. Вкладка, предназначенная для редактирования отношения

В появившейся вкладке пользователь может изменить наименование, выбрать тип и задать соответствующее тело отношения. Кроме того, для отношения можно добавить описание.

Существует четыре типа отношения:

- 1) формула — математическая формула, состоящая из переменных и операторов;
- 2) условное отношение — продукционное отношение вида «если..., то..., иначе...»;
- 3) ограничение — продукционное отношение, накладывающее ограничение на значение параметров;
- 4) сложное отношение — программируемое отношение для опытного пользователя.

Каждому типу отношения соответствует свое тело отношения. Чтобы изменить тело отношения, необходимо нажать на кнопку «...» рядом с соответствующим полем.

5. Руководство по КЭСМИ РАЗУМАТОР WI!MI 2.1

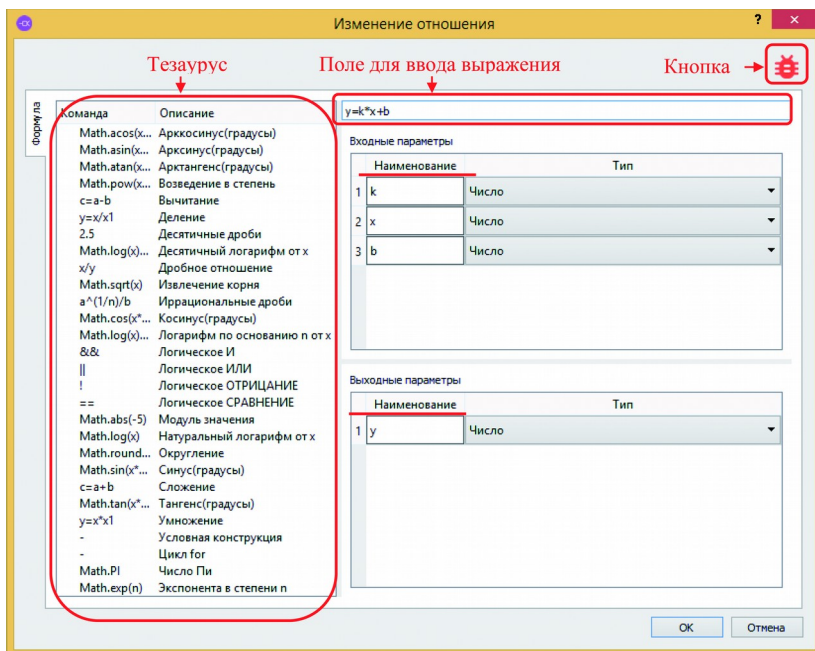


Рисунок 5.9. Диалоговое окно «Изменение отношения» для типа отношения «Формула»

При выборе любого типа отношения в левой части окна отображается тезаурус примеров корректной записи. Можно выбрать отношение оттуда, кликнув по нему два раза, а можно создать свое. Существуют следующие правила создания отношений.

1. Отношение записывается так, чтобы в левой части от знака «равно» стояли выходные переменные, а справа входные.
2. В отношениях можно использовать любые латинские буквы и символы операторов.
3. Переменным типа «строка» значения присваиваются следующим образом: $y = \text{значение строки}$.

4. Переменные в различных отношениях могут быть одинаковыми и абстрактными.
5. В отношении не могут быть одновременно одни и те же входные и выходные переменные. Например, отношение вида $y = y + 1$ недопустимо!

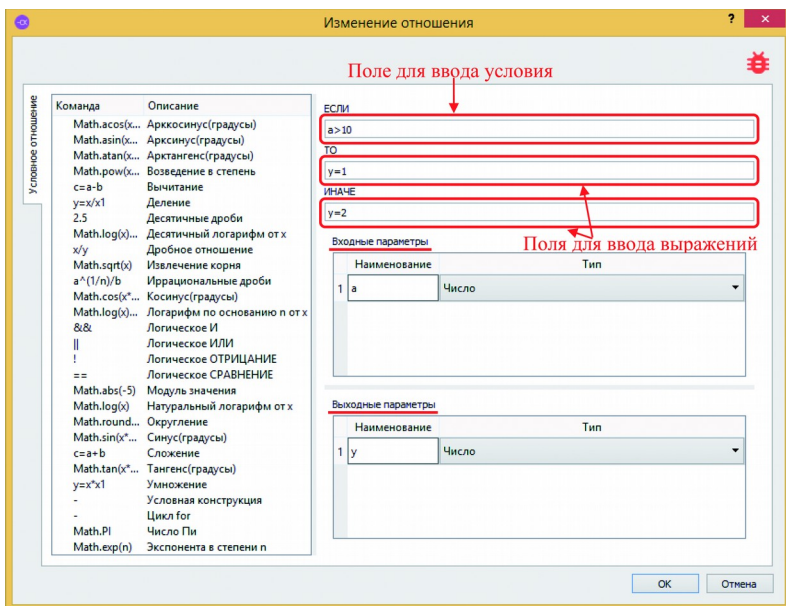


Рисунок 5.10. Изменение отношения типа «Условное отношение»

После создания отношения его необходимо проанализировать, нажав соответствующую кнопку в верхней части диалогового окна. Если тело отношения записано корректно, то программа выделит входные и выходные переменные отношения в соответствующих таблицах диалога. Числовой тип присваивается переменным по умолчанию. В случае необходимости его можно изменить на текстовый. Для завершения создания отношения нажмите на кнопку «ОК». Новое отношение появится в древовидном списке в правой части окна приложения.

5. Руководство по КЭСМИ РАЗУМАТОР WI!MI 2.1

Диалоговое окно «Изменение отношения» имеет разный вид в зависимости от выбранного типа отношения, который указан в левой части этого окна (рис. 5.10).

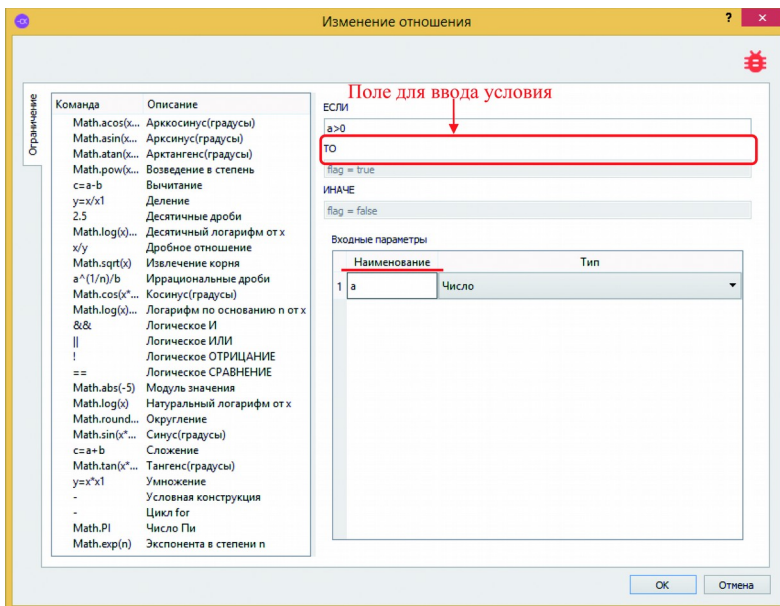


Рисунок 5.11. Изменение отношения типа «Ограничение»

Если выбран условный тип отношения (рис 5.11), то для его настройки используются три поля. В поле «ЕСЛИ» указывается условие выполнения отношения, в поля «ТО» и «ИНАЧЕ» — выражения, выполняемые в случае выполнения и не выполнения условия соответственно.

Аналогичные поля используются и для настройки тела отношения с типом «Ограничение» (рис. 5.11). Для настройки ограничения достаточно указать условие его выполнения в поле «ЕСЛИ».

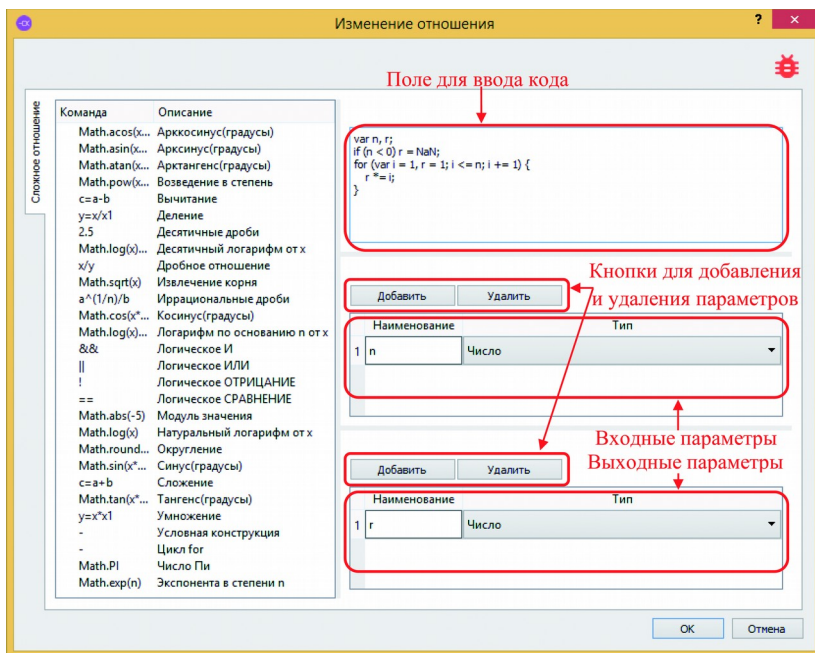


Рисунок 5.12. Создание сложного отношения на примере функции вычисления факториала

Для создания сложного отношения необходимо обладать навыками программирования. Чтобы создать сложное отношение (рис. 5.12), нужно записать код на языке JavaScript в доступное поле, после чего выполнить анализ выражения, нажав на соответствующую кнопку. Если ошибок не обнаружено, то необходимо добавить входные и выходные параметры при помощи соответствующей кнопки. Если были созданы лишние параметры, то их можно удалить. Для этого выделите любое поле строки таблицы, относящейся к удаляемому параметру, и нажмите соответствующую кнопку, расположенную выше таблицы. Входные параметры описываются в верхней таблице, выходные — в нижней. Для определения параметра впишите в поле

«Наименование» соответствующее название переменной из кода.

Созданные отношения открывают возможность создания правил и ограничений на их основе.

5.5.2.1. Правила

Чтобы создать правило по заданному отношению, необходимо щелкнуть правой кнопкой мыши по этому отношению из списка в правой части окна приложения выбрать пункт «Добавить правило» (рис. 5.13). Правило можно создать только если отношение имеет тип «формула», «условное отношение» или «сложное отношение».

В появившейся вкладке (рис. 5.14) пользователь может изменить наименование, добавить описание и выбрать параметры, для которых справедливо правило. Чтобы определить параметры, нажмите на кнопку «...» в соответствующей строке таблицы. В появившемся диалоговом окне выберите параметр из списка существующих и нажмите на кнопку «ОК». С помощью поля над списком можно осуществить поиск параметров по имени. Если параметр был выбран неверно, вы можете очистить поле таблицы, нажав на соответствующую кнопку.

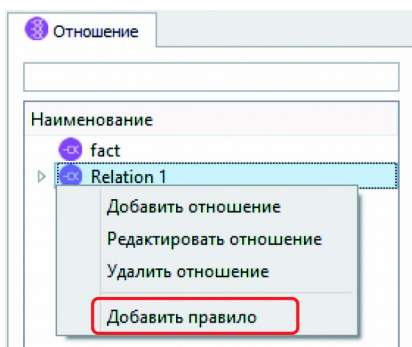


Рисунок 5.13. Контекстное меню, вызываемое для добавления правила

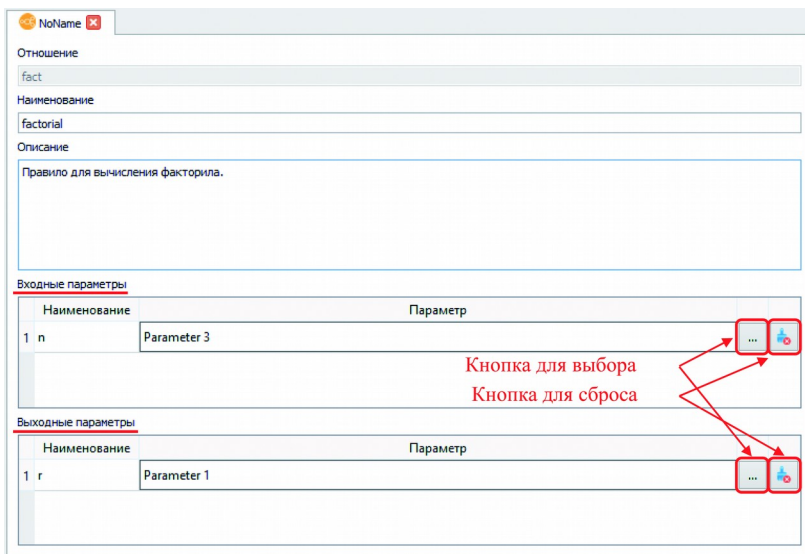


Рисунок 5.14. Вкладка, предназначенная для редактирования правила

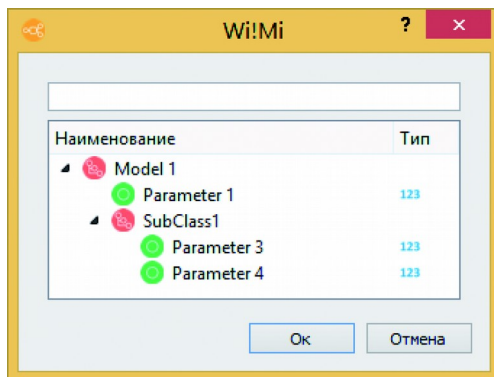


Рисунок 5.15. Диалоговое окно выбора параметров

После заполнения необходимых полей сохраните модель. Новое правило отобразится в списке дочерних элементов отношения, на котором основано правило, в правой части приложения.

Чтобы отредактировать или удалить существующее правило, правым щелчком мыши вызовите контекстное меню (рис. 5.16) и выберите интересующее вас действие.

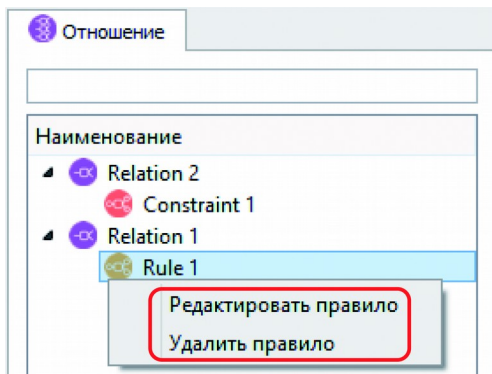


Рисунок 5.16. Контекстное меню, вызываемое по щелчку на правиле

5.5.2.2. Ограничения

Ограничение можно создать только на основе отношения соответствующего типа. Чтобы создать ограничение, необходимо щелкнуть правой кнопкой мыши по отношению из списка в правой части окна приложения и выбрать пункт «Добавить ограничение» (рис. 5.17).

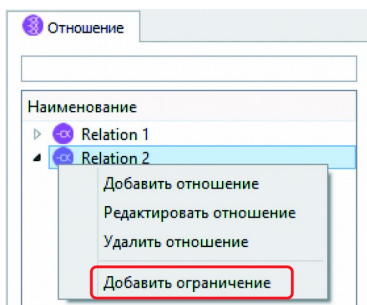


Рисунок 5.17. Контекстное меню, вызываемое по щелчку на отношении с типом «ограничение»

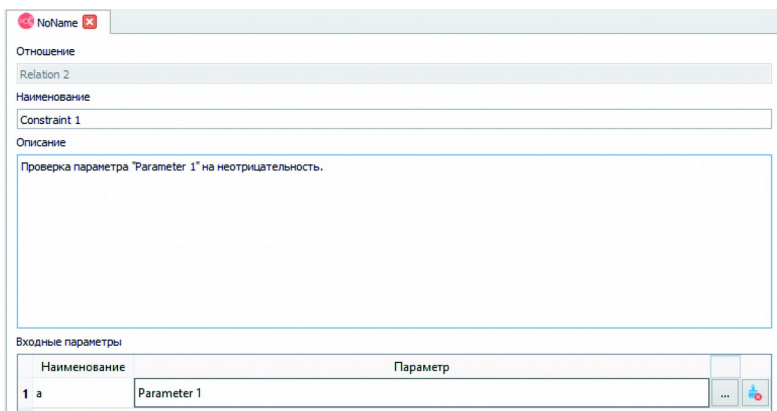


Рисунок 5.18. Вкладка, предназначенная для редактирования ограничения

В появившейся вкладке (рис. 5.18) пользователь может изменить наименование, добавить описание и выбрать параметры, для которых справедливо ограничение. Процесс выбора параметров аналогичен соответствующей процедуре для правил.

После заполнения необходимых полей сохраните модель. Новое ограничение отобразится в списке дочерних элементов отношения, на котором основано правило, в правой части приложения. Чтобы отредактировать или удалить существующее ограничение, правым щелчком мыши вызовите контекстное меню (рис. 5.19) и выберите требуемое действие.

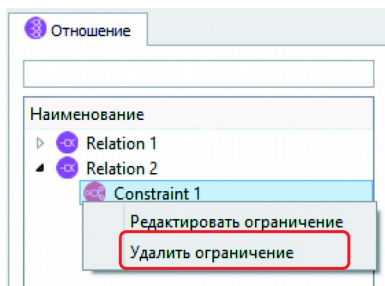


Рисунок 5.19. Контекстное меню для ограничения

Ограничение для конкретного параметра можно создать, щелкнув по кнопке «+» в вкладке редактирования параметра (рис. 5.6). В появившемся окне (рис. 5.20) необходимо указать условия ограничения. Возможно добавление нескольких условий. Кроме того, можно определить связь между условиями («логическое И» или «логическое ИЛИ»).

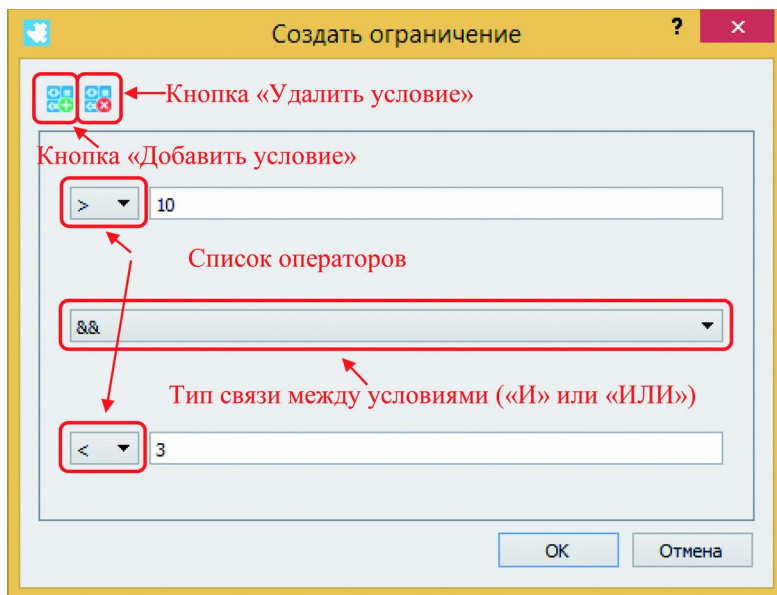


Рисунок 5.20. Окно «Создать ограничение»

Ограничения используются для проверки данных. Во время тестирования модели (см. раздел 5.5.4), если какое-либо ограничение не выполняется, пользователь получает соответствующее сообщение в консоли, а дальнейший расчет прекращается.

5.5.3. Тестирование модели

Тестирование модели предназначено для логического вывода пути от начальных данных к выходным, проверки данных на корректность и расчета необходимых значений. Чтобы осуще-

ствить тестирование, щелкните правой кнопкой мыши на соответствующем классе модели и в контекстном меню выберите пункт «Тестирование модели». Также открыть данную вкладку можно с помощью меню «Инструменты» (рис. 5.21).

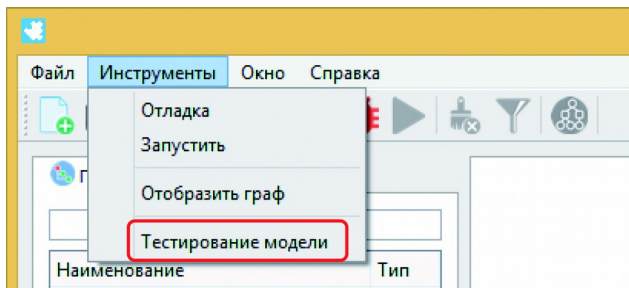


Рисунок 5.21. Меню «Инструменты»

В отображенной вкладке находится таблица, содержащая все параметры, принадлежащие выбранному классу и его потомкам. У каждого из параметров есть поле для ввода исходных данных (значений по умолчанию) и флажок «Найти».

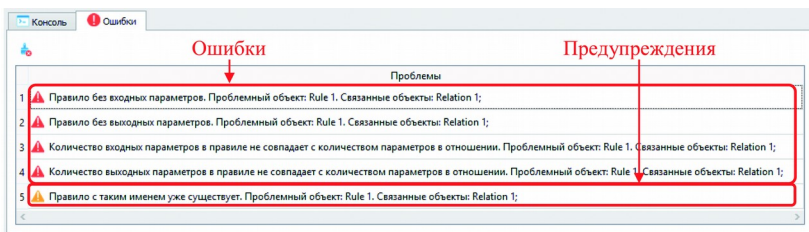


Рисунок 5.22. Вкладка «Ошибки»

Перед началом тестирования имеет смысл проверить модель на ошибки. Для этого нажмите на кнопку «Ошибки», расположенную на панели инструментов. Также выполнить данную операцию можно, выбрав пункт «Отладка» из меню «Инструменты». Если в модели обнаружены проблемы, соответствующие сообщения будут представлены во вкладке «Ошибки»

5. Руководство по КЭСМИ РАЗУМАТОР WIIMI 2.1

(рис. 5.22). Сообщения бывают двух типов: ошибки и предупреждения. В случае возникновения ошибки сохранение модели, а также ее тестирование невозможно до тех пор, пока проблема не будет устранена. При возникновении предупреждения модель останется работоспособной, однако рекомендуется исправлять проблемы, на которые указывает предупреждение, ввиду повышения стабильности модели и уменьшения вероятности появления новых ошибок при дальнейшей работе с ней.

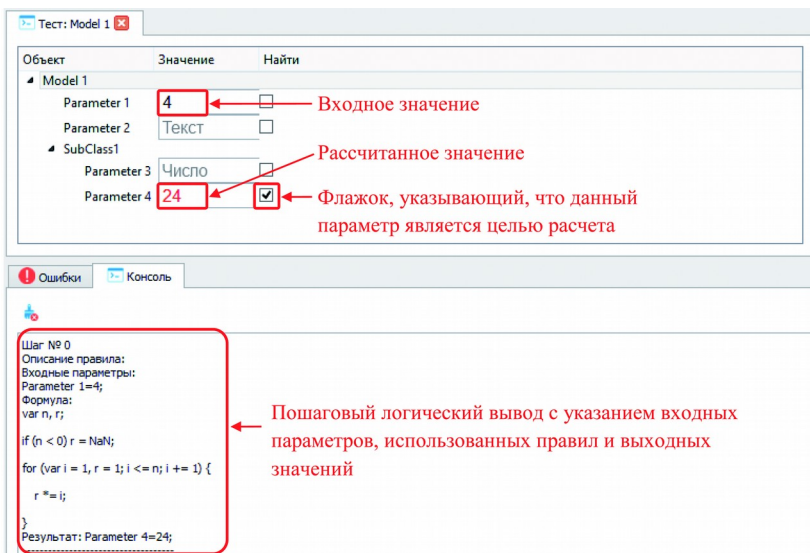


Рисунок 5.23. Тестирование модели

Чтобы протестировать модель (рис. 5.23), требуется ввести исходные данные и отметить флажки у параметров, которые пользователь планирует рассчитать. После чего нужно нажать кнопку «Запустить», расположенную на панели инструментов. Также выполнить данную операцию можно, если выбрать пункт «Запустить» в меню «Инструменты». В случае успеха рассчитанные параметры будут выделены красным цветом, а в консоли

будет выводиться вся текущая информация: сработавшие ограничения (рис. 5.24), найденные и не найденные значения и логический вывод с указанием использованного правила, его описания, входных, выходных и найденных значений.

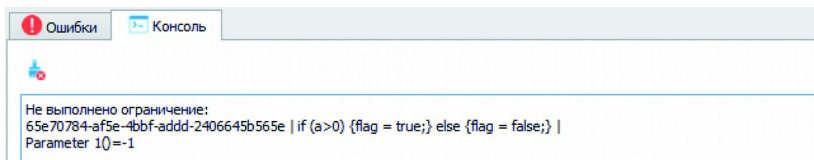


Рисунок 5.24. Вывод консоли при невыполнении ограничения

Перед каждым новым тестированием очищайте результаты старых тестов! В противном случае, новый расчет не будет осуществлен, а приложение напишет сообщение о том, что для поиска заданного параметра не найден путь. Очищать результаты теста можно вручную, а можно использовать соответствующую кнопку на панели инструментов.

После проведения успешного тестирования можно отключить отображение исходных данных. Для этого используйте кнопку «Отобразить только результат», расположенную на панели инструментов (рис. 5.25).



Рисунок 5.25. Кнопки панели инструментов, связанные с тестированием модели

5.5.4. Визуальное представление

Визуально миварная модель приводится в графовом представлении.

Чтобы отобразить граф решения (рис. 5.26), необходимо выполнить тестирование модели. В случае успешного тестирования откройте меню «Инструменты» и выберите пункт «Отобразить граф». Так же это можно сделать, если нажать на соответствующую кнопку на панели инструментов (рис. 5.27).

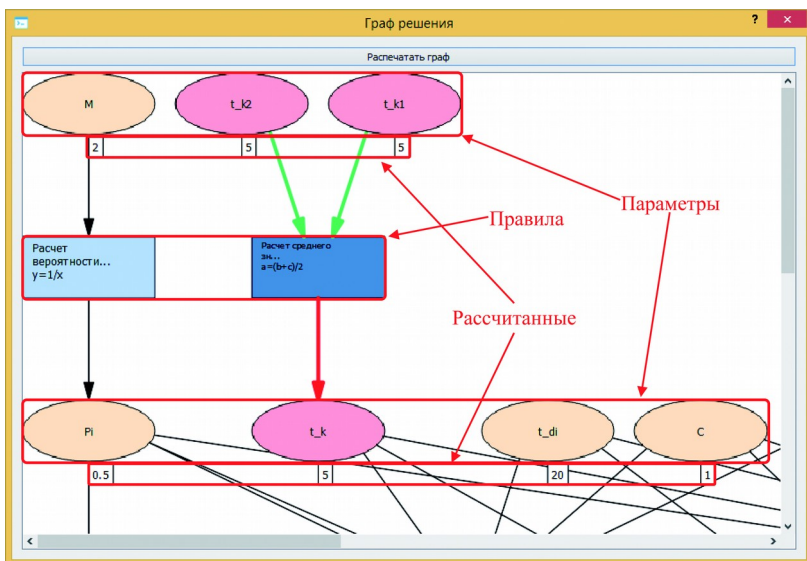


Рисунок 5.26. Граф решения



Рисунок 5.27. Кнопка «Отобразить граф»

Правила обозначаются прямоугольниками, параметры — эллипсами. При щелчке по правилу отображаются все входящие и выходящие параметры, которые соединяются с правилом зелеными и синими стрелками соответственно. Щелчок по параметру показывает все связанные с ним правила и параметры. Около каждого параметра указывается его текущее значение.

5. Руководство по КЭСМИ РАЗУМАТОР WIIMI 2.1

Перемещаться по графу можно с помощью полос прокрутки и мыши. Для перемещения с помощью мыши наведите курсор на пустое место на графе, опустите левую кнопку мыши, и, не отпуская кнопку, перемещайте курсор. Вместе с движением курсора будет перемещаться и граф.

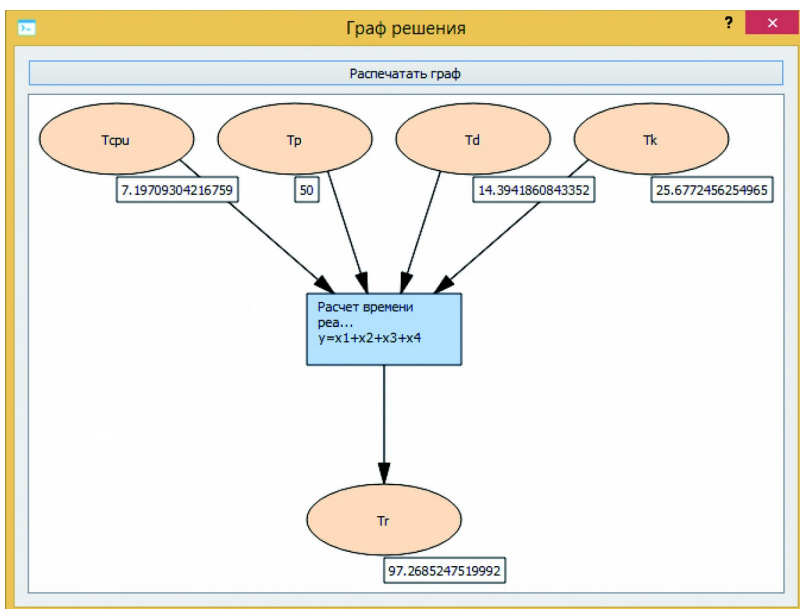


Рисунок 5.28. Увеличенная часть графа, состоящая из выбранного правила и связанных с ним параметров

Если щелкнуть на какое-либо правило дважды, то будет отображена лишь часть графа, состоящая из данного правила и связанных с ним входных и выходных параметров (рис. 5.28). Для возврата к обзору полного графа необходимо снова осуществить двойной щелчок по правилу.

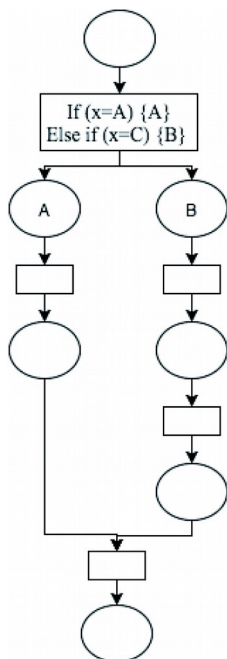


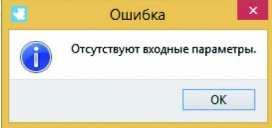
Рисунок 5.29. Иллюстрация технологии runtime

Существует возможность вывода графа на печать. Для этого откройте окно графа и щелкните на кнопке «Распечатать граф». В открывшемся диалоговом окне выберите нужные параметры. По завершению настройки нажмите на кнопку «Печать».

5.5.5. Технология runtime

Технология runtime, добавленная во второй версии Wi!Mi, при построении алгоритма решения задачи учитывает значения рассчитанных промежуточных параметров. Так, при наличии нескольких альтернативных цепочек система основывает свой выбор на значении рассчитанного параметра (см. рис. 5.29). Таким образом, если правила имеют несколько выходных параметров, а рассчитаны будут только некоторые из них, то дальнейший алгоритм может быть перестроен так, чтобы в расчетах участвовали только рассчитанные параметры.

5.6. Возможные ошибки

Описание проблемы	Действия для устранения проблемы	Снимок окна с ошибкой
Не заданы исходные данные	Проверьте, заданы ли исходные данные	
Сообщение «Путь не найден»	Проверьте существующие правила на наличие ошибок; очистите результаты предыдущего тестирования	
Сообщение «Формула не прошла проверку на соответствие данных»	Выполните проверку формулы, используя кнопку «Отладка»	
Сообщение «Ошибка отладки»	Проверьте код на наличие синтаксических ошибок; убедитесь, что все используемые переменные объявлены – входные и выходные параметры тоже должны быть объявлены в коде	
Сообщение «Отсутствуют входные/выходные параметры»	Добавьте соответствующие отношению входные/выходные параметры	
Сообщение «Заполните IF-THEN-ELSE»	Добавьте условие в блок «ЕСЛИ», проверьте выражения в блоках «ТО», «ИНАЧЕ»	

5.7. Рекомендации по освоению

Для более тщательного освоения организованы курсы обучения Wi!Mi. Подробнее ознакомится с проведением курсов и записаться на них можно на официальном сайте компании МИВАР (www.mivar.ru).

6. ЛАБОРАТОРНЫЕ РАБОТЫ

6.1. Описание процесса выполнения лабораторных работ

Цели:

- обучение созданию многомерных баз данных с линейно-вычислительной логикой обработки данных;
- обучение описанию предметных областей с последующим анализом данных, прогнозированием ситуации, созданием модели и последующим расчетом значений.

Проведение лабораторных работ предусматривает два варианта.

1 вариант. Студент имеет возможность описания одной из 30 представленных ниже предметных областей:

- оценка стоимости подзапроса;
- расчет быстродействия элементов ЭВМ;
- 7-й класс алгебра;
- 8-й класс алгебра;
- 9-й класс алгебра;
- прогноз нагрузки работы сервера;
- экономическая оценка инвестиционных проектов;
- поиск квартиры;
- расчет возведения моста;
- диагностика работы ПК;
- профорентация;
- диагностика неисправности работы компьютерной сети;

6. Лабораторные работы

- расчет естественного освещения;
- расчет загазованности автомобильно-дорожного тоннеля;
- подбор вентиляционного оборудования для автомобильно-дорожного тоннеля;
- защита от шума на производстве;
- электробезопасность на производстве;
- методы очистки атмосферного воздуха от загрязнителей;
- безопасность жизнедеятельности (микроклимат);
- методика проведения инвентаризации выбросов загрязняющих веществ в атмосферу для баз дорожной техники (расчетным методом);
- строительство здания;
- электроника (1-й курс);
- электротехника (лабораторные работы);
- комбинаторика;
- школьная химия;
- дифференциальные уравнения;
- воздействие радиации на человека;
- оценка пробеговых выбросов автотранспортных средств в эксплуатации;
- расчет защиты от тепловых излучений;
- оценка возможности возгорания.

Также студент может описать свою предметную область, предварительно согласовав ее с преподавателем.

Любой из предложенных способов подразумевает составление отчета с подробным описанием выбранной предметной области и созданной модели.

2 вариант. Студент составляет модели по предметным областям, описанным в лабораторных работах № 2–5. Выполнение каждой лабораторной работы подтверждается отчетом в бумажном виде.

Задачи к выполнению лабораторных работ по 1 варианту.

- Выбор и подробное описание предметной области. Описание входных и выходных данных модели с последующим представлением их в таблице.
- Добавление классов и параметров модели. Описание их параметров, атрибутов, значений.
- Добавление в модель отношений исходя из выбранных типов: формула, условное отношение, ограничение, сложное.
- Добавление правил с последующим описанием.
- Добавление в модель продукционных отношений, накладывающих ограничения на значения параметров.
- Проверка модели с помощью графового отображения.
- Проверка работоспособности модели с помощью «Умного калькулятора».
- Написание отчета работы модели в виде краткого руководства пользователя.

Задачи к выполнению лабораторных работ по 2 варианту проведения.

- Разработка модели «Физика, 7-й класс».
- Разработка модели «Физика, 8-й класс».
- Разработка модели «Физика, 9-й класс».
- Разработка модели «Генетика».

6. Лабораторные работы

- Разработка модели «Аналитическое моделирование РСОД методом фонового потока».

По задачам 1–4 необходимо описать предметную область модели и произвести расчеты согласно представленным задачам. В задаче 5 необходимо также описать предметную область модели, но расчеты нужно произвести согласно представленным параметрам.

6.2. Лабораторная работа № 1

Модель «Физика 7-й класс»

Описание

Необходимо создать модель, в которой будут использоваться классы, параметры, отношения и правила, указанные ниже. Работоспособность модели следует проверить, решив задачи из школьной программы по физике.

Задание 1

Определите следующие классы:

- взаимодействие тел;
- давление твердых тел, жидкостей и газов;
- работа, мощность, энергия.

Задание 2

В класс «Взаимодействие тел» добавьте параметры из табл. 6.2.1.

Таблица 6.2.1. Параметры класса «Взаимодействие тел»

Обозначение	Тип параметра	Описание
v	Число	Скорость [м/с]
s	Число	Путь [м]

6. Лабораторные работы

t	Число	Время [с]
ρ	Число	Плотность [кг/м ³]
V	Число	Объем [м ³]
$F_{\text{Тяж}}$	Число	Сила тяжести [Н]
g	Число	10 м/с ² — ускорение свободного падения
m	Число	Масса [кг]
R	Число	Равнодействующая [Н]
F_1	Число	1-я сила [Н]
F_2	Число	2-я сила [Н]
F_3	Число	3-я сила [Н]
F_4	Число	4-я сила [Н]
P	Число	Вес тела [Н]
$F_{\text{упр}}$	Число	Сила упругости [Н]
k	Число	Жесткость [Н/м]
Δl	Число	Удлинение [м]

Задание 3

В класс «Давление твердых тел, жидкостей и газов» добавьте параметры из табл. 6.2.2.

Таблица 6.2.2. Параметры класса «Давление твердых тел, жидкостей и газов»

Обозначение	Тип параметра	Описание
F	Число	Сила при давлении [Н]
S	Число	Площадь [м ²]
p	Число	Давление [Па]
h	Число	Высота жидкости [м]
$F_{\text{Арх}}$	Число	Сила Архимеда [Н]
ρ	Число	Плотность [кг/м ³]
V	Число	Объем [м ³]
m	Число	Масса [кг]
g	Число	10 м/с ² — ускорение свободного падения

Задание 4

В класс «Работа, мощность, энергия» добавьте параметры из таблицы 6.2.3.

6. Лабораторные работы

Таблица 6.2.3. Параметры класса «Работа, мощность, энергия»

Обозначение	Тип параметра	Описание
A	Число	Работа [Дж]
F	Число	Сила [Н]
N	Число	Мощность [Вт]
l	Число	Плечо силы [м]
l_1	Число	1-е плечо силы [м]
l_2	Число	2-е плечо силы [м]
M	Число	Момент силы [Н·м]
η	Число	Коэффициент полезного действия [%]
A_n	Число	Полезная работа [Дж]
$A_з$	Число	Затраченная работа [Дж]
E_n	Число	Потенциальная энергия [Дж]
E_k	Число	Кинетическая энергия [Дж]
S	Число	Перемещение тела [м]
T	Число	Время выполнения работы [с]
M	Число	Масса тела [кг]
h	Число	Высота [м]
v	Число	Скорость [м/с]
ρ	Число	Плотность [кг/м ³]
V	Число	Объем [м ³]
$F_{Тяж}$	Число	Сила тяжести [Н]
F_1	Число	Сила, действующая на 1-е плечо
F_2	Число	Сила, действующая на 2-е плечо

Задание 5

Добавьте отношения и правила, которые будут реализовывать следующие формулы.

Для класса «Взаимодействие тел»:

$$v = \frac{s}{t};$$

$$t = \frac{s}{v};$$

$$s = vt;$$

$$m = \frac{F_{Тяж}}{g};$$

$$R = |(F_1 + F_2) - (F_3 + F_4)|;$$

$$P = gm;$$

6. Лабораторные работы

$$\begin{array}{ll} \rho = \frac{m}{V}; & m = \frac{P}{g}; \\ V = \frac{m}{\rho}; & F_{\text{упр}} = k \Delta l; \\ m = \rho V; & k = \frac{F_{\text{упр}}}{\Delta l}; \\ F_{\text{Тяж}} = gm; & \Delta l = \frac{F_{\text{упр}}}{k}. \end{array}$$

Для класса «Давление твердых тел, жидкостей и газов»:

$$\begin{array}{ll} p = \frac{F}{S}; & h = \frac{p}{\rho g}; \\ F = pS; & F_{\text{Аpx}} = g \rho V; \\ S = \frac{F}{p}; & \rho = \frac{F_{\text{Аpx}}}{gV}; \\ p = \rho gh; & V = \frac{F_{\text{Аpx}}}{g \rho}; \\ \rho = p \frac{p}{gh}; & \end{array}$$

Для класса «Работа, мощность, энергия»:

$$\begin{array}{ll} A = FS; & A_{\text{н}} = \frac{\eta A_3}{100}; \\ F = \frac{A}{S}; & A_3 = \frac{A_{\text{н}} 100}{\eta}; \\ S = \frac{A}{F}; & E_{\text{н}} = mgh; \\ N = \frac{A}{T}; & m = \frac{E_{\text{н}}}{gh}; \\ A = NT; & h = \frac{E_{\text{н}}}{mg}; \\ T = \frac{A}{N}; & E_{\text{к}} = \frac{m v^2}{2}; \\ l_1 = \frac{F_2 l_2}{F_1}; & m = \frac{2 E_{\text{к}}}{v^2}; \\ l_2 = \frac{F_1 \cdot l_1}{F_2}; & \end{array}$$

6. Лабораторные работы

$$F_1 = \frac{F_2 l_2}{l_1};$$

$$F_2 = \frac{F_1 l_1}{l_2};$$

$$M = Fl;$$

$$\eta = \frac{A_n}{A_3} 100;$$

$$v = \sqrt{\frac{2 E_k}{m}};$$

$$F = mg;$$

$$m = \rho v;$$

$$F_2 = mg.$$

Задание 6

Проверьте работу модели, решив задачи, приведенные ниже. Список задач индивидуален для каждого варианта и определяется по табл. 6.2.4.

Таблица 6.2.4. Варианты для решения задач

№ задачи	Вариант 1	Вариант 2	Вариант 3	Вариант 4	Вариант 5	Вариант 6	Вариант 7	Вариант 8	Вариант 9	Вариант 10	Вариант 11	Вариант 12	Вариант 13	Вариант 14	Вариант 15	Вариант 16	Вариант 17	Вариант 18	Вариант 19	Вариант 20
1	+			+		+		+	+		+		+	+		+		+	+	
2	+	+			+		+		+	+		+		+	+		+		+	+
3		+	+			+		+		+	+		+		+	+		+		+
4	+		+	+			+		+		+	+		+		+	+		+	
5		+		+	+			+		+		+	+		+		+	+		+
6	+		+		+	+			+		+		+	+		+		+	+	
7	+	+		+		+	+			+		+		+	+		+		+	+
8		+	+		+		+	+			+		+		+	+		+		+
9	+		+	+		+		+	+			+		+		+	+		+	
10		+		+	+		+		+	+			+		+		+	+		+
11	+		+		+	+		+		+	+			+		+		+	+	
12	+	+		+		+	+		+		+	+			+		+		+	+

6. Лабораторные работы

13		+	+		+		+	+		+		+	+			+		+		+
14	+		+	+		+		+	+		+		+	+			+		+	
15		+		+	+		+		+	+		+		+	+			+		+
16	+		+		+	+		+		+	+		+		+	+			+	
17		+		+		+	+		+		+	+		+		+	+			+
18			+		+		+	+		+		+	+		+	+		+		+

Типовые задачи

Задача 1

Поезд, двигаясь равномерно, за 2 ч проходит путь, равный 108 км. Вычислите скорость движения поезда. ($v = \frac{s}{t}$) (15 м/с).

Задача 2

Самолет летит со средней скоростью 750 км/ч. Какой путь он пролетит за 360 с? ($s = vt$) (75 км).

Задача 3

Самое легкое дерево — бальза. Масса древесины этого дерева равна 12 г при объеме 100 см³. Определите плотность древесины. ($\rho = \frac{m}{V}$) (120 кг/м³).

Задача 4

На столе стоит чайник с водой массой 1,5 кг. Определите силу тяжести и вес чайника. ($F_{\text{тяж}} = gm$, $P = gm$) (15 Н).

Задача 5

В игре по перетягиванию каната участвуют 4 человека. Два из них тянут канат в одну сторону с силами 300 Н и 400 Н, два — в противоположную сторону с силами 330 Н и 380 Н. Чему равна равнодействующая этих сил? ($R = |(F_1 + F_2) - (F_3 + F_4)|$) (10 Н).

6. Лабораторные работы

Задача 6

Объем бензина в баке автомобиля во время поездки уменьшился на 25 л. Плотность бензина равна 710 кг/м^3 . Насколько уменьшился вес автомобиля? ($m = \rho V$; $P = mg$) ($177,5 \text{ Н}$).

Задача 7

Какие силы надо приложить к концам проволоки, жесткость которой 100 кН/м , чтобы растянуть ее на 1 мм ? ($F_{\text{упр}} = k \Delta l$) (100 Н).

Задача 8

Какое давление создает на фундамент кирпичная стена высотой 10 м ? Плотность кирпича равна 1600 кг/м^3 . ($p = \rho gh$) (160 КПа).

Задача 9

Объем куска железа $0,1 \text{ дм}^3$. Плотность керосина 800 кг/м^3 . Какая выталкивающая сила будет на него действовать при полном погружении в керосин? ($F_{\text{Арх}} = g \rho V$) ($0,8 \text{ Н}$).

Задача 10

При массе 60 кг и росте $1,6 \text{ м}$ площадь поверхности тела человека равна $\approx 1,6 \text{ м}^2$. Рассчитайте силу, с которой атмосфера давит на человека (при нормальном атмосферном давлении — $101\,325 \text{ Па}$). ($F = pS$) (162 кН).

Задача 11

При помощи лебедки равномерно поднят груз на высоту 10 м в течение $0,8 \text{ мин}$, причем совершена работа, равная 120 кДж . Вычислите силу для поднятия груза и развиваемую при его подъеме мощность. ($F = \frac{A}{S}$; $N = \frac{A}{T}$) ($12\,000 \text{ Н}$; $2,5 \text{ кВт}$).

Задача 12

Вычислите работу, совершаемую при подъеме гранитной плиты объемом $0,5 \text{ м}^3$ на высоту 20 м . Плотность гранита 2500 кг/м^3 . ($F = mg$; $A = FS$; $m = \rho V$) (250 кДж).

6. Лабораторные работы

Задача 13

Найти мощность потока воды, протекающей через плотину, если высота падения воды 25 м, а расход — 120 м³ в минуту.

$$(m = \rho V; F = mg; A = FS; N = \frac{A}{T}) (0,5 \text{ МВт}).$$

Задача 14

С помощью рычага рабочий поднимает плиту массой 240 кг. Какую силу прикладывает он к большему плечу рычага, равному

$$2,4 \text{ м, если меньшее плечо равно } 0,6 \text{ м? } (F_1 = \frac{F_2 l_2}{l_1}; F_2 = mg.)$$

(600 Н).

Задача 15

Найти момент силы, если плечо равно 4 м, а сила оказываемая на него равна 5 Н? ($M = Fl$) (20 Н·м).

Задача 16

При движении в гору велосипедистом была совершена работа в 8 кДж. КПД велосипедиста равен 75%. Найдите полезную

$$\text{работу. } (A_n = \frac{\eta A_s}{100}) (6 \text{ кДж}).$$

Задача 17

Какова кинетическая энергия космического корабля «Союз» массой 6,6 т, движущегося по орбите со скоростью 7,8 км/с?

$$(E_k = \frac{m v^2}{2}) (2 \cdot 10^{11} \text{ Дж}).$$

Задача 18

На какой высоте потенциальная энергия груза массой 2 т равна

$$10 \text{ кДж? } (h = \frac{E_n}{mg}) (0,5 \text{ м}).$$

6. Лабораторные работы

Задание 7

Сделать отчет по плану о выполнении лабораторной работы.

Задание 8

Подготовить ответы на вопросы по теоретическому материалу. Защитить выполнение лабораторной работы:

- 1) показать работающую модель;
- 2) представить отчет в бумажном виде.

6.3. Лабораторная работа № 2

Модель «Физика 8-й класс»

Описание

Необходимо создать модель, в которой будут использоваться классы, параметры, отношения и правила, указанные ниже. Работоспособность модели следует проверить, решив задачи школьной программы по физике.

Задание 1

Определите следующие классы:

- тепловые явления;
- электрические и электромагнитные явления;
- световые явления.

Задание 2

В класс «Тепловые явления» добавьте параметры из табл. 6.3.1.

Таблица 6.3.1. Параметры класса «Тепловые явления»

Обозначение	Тип параметра	Описание
Q	Число	Количество теплоты [Дж]
m	Число	Масса [кг]
Δt	Число	$(t_2 - t_1)$ — разность температур [°C]

6. Лабораторные работы

t_1	Число	Начальная температура [°С]
t_2	Число	Конечная температура [°С]
C	Число	Удельная теплоемкость [Дж/кг. °С]
q	Число	Удельная теплота сгорания топлива [Дж/кг]
λ	Число	Удельная теплота плавления [Дж/кг]
L	Число	Удельная теплота парообразования [Дж/кг]
$Q_{\text{пл}}$	Число	Количество теплоты при плавлении [Дж]
$Q_{\text{нг}}$	Число	Количество теплоты при нагревании [Дж]
$Q_{\text{сг}}$	Число	Количество теплоты при сгорании [Дж]
$Q_{\text{пар}}$	Число	Количество теплоты при парообразовании [Дж]
V	Число	Объем тела/жидкости/газа [м ³]
ρ	Число	Плотность тела/жидкости/газа [кг/м ³]
η	Число	Коэффициент полезного действия [%]
$A_{\text{п}}$	Число	Полезная работа [Дж]
$A_{\text{з}}$	Число	Затраченная работа [Дж]

Задание 3

В класс «Электрические явления» добавьте параметры из табл. 6.3.2.

Таблица 6.3.2. Параметры класса «Электрические явления»

Обозначение	Тип параметра	Описание
I	Число	Сила тока [А]
q	Число	Электрический заряд [Кл]
T	Число	Время [с]
U	Число	Напряжение [В]
A	Число	Работа [Дж]
R	Число	Сопротивление [Ом]
ρ	Число	Удельное сопротивление [Ом·мм ² /м]
Δl	Число	Длина [м]
s	Число	Площадь поперечного сечения [мм ²]
$A_{\text{т}}$	Число	Работа тока [Дж]
P	Число	Мощность [Вт]
$Q_{\text{т}}$	Число	Количество теплоты, выделяемое проводником с током [Дж]

6. Лабораторные работы

Задание 4

В класс «Световые явления» добавьте параметры из табл. 6.3.3.

Таблица 6.3.3. Параметры класса «Световые явления»

Обозначение	Тип параметра	Описание
D	Число	Оптическая сила линзы [дптр]
F	Число	Фокусное расстояние линзы [м]
d	Число	Расстояние от предмета до линзы [м]
f	Число	Расстояние от линзы до изображения [м]

Задание 5

Добавьте отношения и правила, которые будут реализовывать следующие формулы.

Для класса «Тепловые явления»:

$$Q_{\text{нр}} = cm(t_2 - t_1);$$

$$c = \frac{Q_{\text{нр}}}{m(t_2 - t_1)};$$

$$m = \frac{Q_{\text{нр}}}{c(t_2 - t_1)};$$

$$Q_{\text{нр}} = cm \Delta t;$$

$$c = \frac{Q_{\text{нр}}}{m \Delta t};$$

$$m = \frac{Q_{\text{нр}}}{c \Delta t};$$

$$Q = Q_{\text{пл}} + Q_{\text{нр}};$$

$$Q_{\text{ср}} = qm;$$

$$q = \frac{Q_{\text{ср}}}{m};$$

$$m = \frac{Q_{\text{ср}}}{q};$$

$$Q_{\text{пл}} = \lambda m;$$

$$\lambda = \frac{Q_{\text{пл}}}{m};$$

$$m = \frac{Q_{\text{пл}}}{\lambda};$$

$$V = \frac{m}{\rho};$$

$$\rho = \frac{m}{V};$$

$$m = V \rho;$$

$$Q_{\text{пар}} = Lm;$$

$$L = \frac{Q_{\text{пар}}}{m};$$

$$m = \frac{Q_{\text{пар}}}{L};$$

$$\eta = \frac{A_{\text{п}}}{A_3} 100;$$

$$A_3 = mq.$$

6. Лабораторные работы

Для класса «Электрические явления»:

$$\begin{array}{ll} I = \frac{q}{t}; & A_I = UI t; \\ t = \frac{q}{I}; & U = \frac{A_I}{It}; \\ q = It; & I = \frac{A_I}{Ut}; \\ U = \frac{A_I}{q}; & t = \frac{A_I}{UI}; \\ q = \frac{A_I}{U}; & P = UI; \\ U = IR; & U = \frac{P}{I}; \\ I = \frac{U}{R}; & I = \frac{P}{U}; \\ R = \frac{U}{I}; & Q_I = I^2 R t; \\ R = \frac{\rho \Delta l}{s}; & R = \frac{Q_I}{I^2 t}; \\ \rho = \frac{Rs}{\Delta l}; & t = \frac{Q_I}{I^2 R}; \\ \Delta l = \frac{Rs}{\rho}; & I = \sqrt{\frac{Q_I}{Rt}}. \\ s = \frac{\rho \Delta l}{R}; & \end{array}$$

Для класса «Световые явления»:

$$\begin{array}{ll} D = \frac{1}{F}; & D = \frac{d+f}{d f}; \\ F = \frac{1}{D}; & F = \frac{d f}{d+f}. \end{array}$$

Задание 6

Проверьте работу модели, решив задачи, приведенные ниже. Список задач индивидуален для каждого варианта и определяется по табл. 6.3.4.

6. Лабораторные работы

Таблица 6.3.4. Варианты для решения задач

№ задачи	Вариант 1	Вариант 2	Вариант 3	Вариант 4	Вариант 5	Вариант 6	Вариант 7	Вариант 8	Вариант 9	Вариант 10	Вариант 11	Вариант 12	Вариант 13	Вариант 14	Вариант 15	Вариант 16	Вариант 17	Вариант 18	Вариант 19	Вариант 20
1	+			+		+		+	+		+		+	+		+		+	+	
2	+	+			+		+		+	+		+		+	+		+		+	+
3		+	+			+		+		+	+		+		+	+		+		+
4	+		+	+			+		+		+	+		+		+	+		+	
5		+		+	+			+		+		+	+		+		+	+		+
6	+		+		+	+			+		+		+	+		+		+	+	
7	+	+		+		+	+			+		+		+	+		+		+	+
8		+	+		+		+	+			+		+		+	+		+		+
9	+		+	+		+		+	+			+		+		+	+		+	
10		+		+	+		+		+	+			+		+		+	+		+
11	+		+		+	+		+		+	+			+		+		+	+	
12	+	+		+		+	+		+		+	+			+		+		+	+

Типовые задачи

Задача 1

Какое количество теплоты нужно затратить, чтобы превратить 2 кг льда, взятого при температуре 0°C , в воду температурой 40°C ? Удельная теплоемкость воды — $4200 \text{ Дж/кг}\cdot^{\circ}\text{C}$, удельная теплота плавления — $3,4\cdot 10^5 \text{ Дж/кг}$. ($Q = Q_{\text{пл}} + Q_{\text{гр}}$; $Q_{\text{гр}} = cm(t_2 - t_1)$; $Q_{\text{пл}} = \lambda m$) (1016 кДж).

Задача 2

Для нагревания воды в баке нужно затратить $4,2\cdot 10^7 \text{ Дж}$ энергии. Удельная теплота сгорания угля $3,4\cdot 10^7 \text{ Дж/кг}$. Сколько для этой цели нужно сжечь древесного угля? ($m = \frac{Q_{\text{ср}}}{q}$) ($1,24 \text{ кг}$).

6. Лабораторные работы

Задача 3

Какое количество теплоты выделилось при остывании воды, объем которой 20 л, если температура изменилась от 100°C до 50°C? Удельная теплоемкость воды 4 200 Дж/кг·°C, плотность воды 1 000 кг/м³. ($m = V \rho$; $Q_{\text{от}} = cm(t_2 - t_1)$) (4200 кДж).

Задача 4

Определите КПД двигателя трактора, которому для выполнения работы 1,89·10⁷ Дж потребовалось 1,5 кг топлива с удельной теплотой сгорания 4,2·10⁷ Дж/кг. ($\eta = \frac{A_{\text{п}}}{A_3} 100$; $A_3 = mq$) (30%).

Задача 5

Сила тока в цепи электрической плитки равна 1,4 А. Какой электрический заряд проходит через поперечное сечение ее спирали за 10 мин? ($q = It$) (840 Кл).

Задача 6

Сила тока в железном проводнике длиной 150 мм и площадью поперечного сечения 0,02 мм² равна 250 мА. Какое напряжение на концах проводника? (Удельное сопротивление проводника – 0,098 Ом·мм²/м) ($U = IR$; $R = \frac{\rho \Delta l}{S}$) (0,18 В).

Задача 7

Рассчитайте сопротивление проводника, изготовленного из алюминиевой проволоки длиной 80 см и площадью поперечного сечения 0,2 мм². (Удельное сопротивление проводника – 0,028 Ом·мм²/м) ($R = \frac{\rho \Delta l}{S}$) (0,112 Ом).

Задача 8

В цепь с напряжением 127 В включена электрическая лампа, сила тока в которой 0,6 А. Найдите мощность тока в лампе. ($P = UI$) (76,2 Вт).

Задача 9

Какое количество теплоты выделится за 30 мин проволочной спиралью сопротивлением 20 Ом при силе тока 5 А? ($Q_I = I^2 R t$) (900 кДж).

Задача 10

Оптическая сила линзы равна 1,6 дптр. Каково фокусное расстояние этой линзы? ($F = \frac{1}{D}$) (0,625 дптр).

Задача 11

Фокусное расстояние линзы 10 см. Чему равна оптическая сила этой линзы? ($D = \frac{1}{F}$) (10 дптр).

Задача 12

Расстояние от предмета до тонкой линзы равно 5 м, а расстояние от линзы до изображения равно 20 м. Чему равен фокус линзы и ее оптическая сила? ($D = \frac{d+f}{d f}$; $F = \frac{1}{D}$) (0,25 дптр; 4 м).

Задание 7

Сделать отчет по плану о выполнении лабораторной работы. План выполнения отчета находится в файле «Отчет по лабораторным работам».

Задание 8

Подготовить ответы на вопросы по теоретическому материалу. Защитить выполнение лабораторной работы:

- 1) показать работающую модель;
- 2) представить отчет в бумажном виде.

6.4. Лабораторная работа № 3

Модель «Физика 9-й класс»

Описание

Необходимо создать модель, в которой будут использоваться классы, параметры, отношения и правила, указанные ниже. Работоспособность модели следует проверить, решив задачи из школьной программы по физике.

Задание 1

Определите следующие классы:

- законы взаимодействия и движения тел;
- электромагнитное поле.

Задание 2

В класс «Законы взаимодействия и движения тел» добавьте подклассы:

- равномерное движение тела;
- равноускоренное движение тела;
- законы Ньютона;
- высота;
- закон всемирного тяготения;
- импульс тела;
- периодические колебания.

Задание 3

В подкласс «Равномерное движение тела» добавьте параметры из табл. 6.4.1.

6. Лабораторные работы

Таблица 6.4.1. Параметры класса «Равномерное движение тела»

Обозначение	Тип параметра	Описание
x_0	Число	Начальная координата по оси Ox [м]
x	Число	Конечная координата по оси Ox [м]
S_x	Число	Проекция вектора перемещения на ось Ox [м]
v_x	Число	Скорость тела [м/с]
t	Число	Время [с]

Задание 4

В подкласс «Равноускоренное движение тела» добавьте параметры из табл. 6.4.2.

Таблица 6.4.2. Параметры класса «Равноускоренное движение тела»

Обозначение	Тип параметра	Описание
v	Число	Скорость [м/с]
v_0	Число	Начальная скорость [м/с]
Δt	Число	Промежуток времени [с]
a	Число	Ускорение [м/с ²]
$a_{ц}$	Число	Центростремительное ускорение [м/с ²]
r	Число	Радиус окружности [м]
t_0	Число	Начальное время [с]
t	Число	Время [с]
x	Число	Координата [м]
x_0	Число	Начальная координата [м]
S	Число	Перемещение [м]

Задание 5

В подкласс «Законы Ньютона» добавьте параметры из табл. 6.4.3.

Таблица 6.4.3. Параметры класса «Законы Ньютона»

Обозначение	Тип параметра	Описание
m	Число	Масса тела [кг]
F	Число	Результирующая [Н]
a	Число	Ускорение [м/с ²]

6. Лабораторные работы

Задание 6

В подкласс «Высота» добавьте параметры из табл. 6.4.4.

Таблица 6.4.4. Параметры класса «Высота»

Обозначение	Тип параметра	Описание
h	Число	Высота [м]
g	Число	Ускорение свободного падения [м/с ²] – $g=9,8$ м/с ²
v_0	Число	Начальная скорость [м/с]
v	Число	Скорость тела [м/с]
t	Число	Время [с]

Задание 7

В подкласс «Закон всемирного тяготения» добавьте параметры из табл. 6.4.5.

Таблица 6.4.5. Параметры класса «Закон всемирного тяготения»

Обозначение	Тип параметра	Описание
$F_{\text{тяг}}$	Число	Сила всемирного тяготения [Н]
m_1	Число	Масса тела №1 [кг]
m_2	Число	Масса тела №2 [кг]
G	Число	Гравитационная постоянная – $G=6,67 \cdot 10^{-11}$ Н·м ² / [Кг] ²
r	Число	Расстояние между телами [м]

Задание 8

В подкласс «Импульс тела» добавьте параметры из табл. 6.4.6.

Таблица 6.4.6. Параметры класса «Импульс тела»

Обозначение	Тип параметра	Описание
p	Число	Импульс [кг·м/с]
m	Число	Масса тела [кг]
v	Число	Скорость тела [м/с]

6. Лабораторные работы

Задание 9

В подкласс «Периодические колебания» добавьте параметры из табл. 6.4.7.

Таблица 6.4.7. Параметры класса «Периодические колебания»

Обозначение	Тип параметра	Описание
T	Число	Период колебаний [с]
ν	Число	Частота колебаний [1/с]
v	Число	Скорость волны [м/с]
l	Число	Длина волны [м]

Задание 10

В класс «Электромагнитное поле» добавьте подклассы:

- индукция магнитного поля;
- энергия магнитного поля;
- энергия электрического поля.

Задание 11

В подкласс «Индукция магнитного поля» добавьте параметры из табл. 6.4.8.

Таблица 6.4.8. Параметры класса «Индукция магнитного поля»

Обозначение	Тип параметра	Описание
B	Число	Индукция [Тл]
F	Число	Сила [Н]
I	Число	Сила тока [А]
l	Число	Длина проводника [м]

Задание 12

В подкласс «Энергия магнитного поля» добавьте параметры из табл. 6.4.9.

6. Лабораторные работы

Таблица 6.4.9. Параметры класса «Энергия магнитного поля»

Обозначение	Тип параметра	Описание
W	Число	Энергия магнитного поля [Дж]
L	Число	Индуктивность [Гн]
I	Число	Сила тока [А]

Задание 13

В подкласс «Энергия электрического поля» добавьте параметры из табл. 6.4.10.

Таблица 6.4.10. Параметры класса «Энергия электрического поля»

Обозначение	Тип параметра	Описание
C	Число	Емкость [Ф]
q	Число	Электрический заряд [Кл]
U	Число	Напряжение [В]
$E_{эл}$	Число	Энергия электрического поля [Дж]

Задание 14

Добавьте отношения и правила, которые будут реализовывать следующие формулы.

Для подкласса «Равномерное движение тела»:

$$\begin{aligned} S_x &= x - x_0 ; & S_x &= v_x t ; \\ t &= \frac{S_x}{v_x} ; & x &= x_0 + v_x t ; \\ v_x &= \frac{S_x}{t} ; & x_0 &= x - v_x t ; \\ & & t &= \frac{x - x_0}{v_x} . \end{aligned}$$

Для подкласса «Равноускоренное движение тела»:

$$\begin{aligned} a &= \frac{v - v_0}{\Delta t} ; & S_x &= \frac{v^2 - v_0^2}{2a} ; \\ \Delta t &= t - t_0 ; \end{aligned}$$

6. Лабораторные работы

$$\begin{aligned}v &= v_0 + a \Delta t ; & x &= x_0 + v_0 t + \frac{a \Delta t^2}{2} ; \\v_0 &= v - a \Delta t ; & a_{\text{ц}} &= \frac{v^2}{r} . \\S_x &= v_0 \Delta t + \frac{a \cdot \Delta t^2}{2} ;\end{aligned}$$

Для подкласса «Законы Ньютона»:

$$\begin{aligned}F &= ma ; & a &= \frac{F}{m} . \\m &= \frac{F}{a} ;\end{aligned}$$

Для подкласса «Высота»:

$$\begin{aligned}h &= v_0 \cdot t + \frac{g t^2}{2} ; & v_0 &= v - g t ; \\v &= v_0 + g t ; & t &= \frac{v - v_0}{g} .\end{aligned}$$

Для подкласса «Закон всемирного тяготения»:

$$\begin{aligned}F_{\text{тяг}} &= G \frac{m_1 m_2}{r^2} ; & m_1 &= \frac{F_{\text{тяг}} r^2}{G m_2} ; \\r &= \sqrt{G \frac{m_1 m_2}{F_{\text{тяг}}}} ; & m_2 &= \frac{F_{\text{тяг}} r^2}{G m_1} .\end{aligned}$$

Для подкласса «Импульс тела»:

$$\begin{aligned}p &= m v ; & v &= \frac{p}{m} . \\m &= \frac{p}{v} ;\end{aligned}$$

Для подкласса «Периодические колебания»:

$$\begin{aligned}T &= \frac{l}{V} ; & v &= \frac{l}{T} ; \\V &= \frac{l}{T} ; & l &= v T ; \\ & & T &= \frac{l}{v} .\end{aligned}$$

6. Лабораторные работы

Для подкласса «Индукция магнитного поля»:

$$\begin{aligned} B &= \frac{F}{Il}; & I &= \frac{F}{Bl}; \\ F &= BIl; & l &= \frac{F}{BI}. \end{aligned}$$

Для подкласса «Энергия магнитного поля»:

$$\begin{aligned} W &= \frac{LI^2}{2}; & I &= \sqrt{\frac{2W}{L}}. \\ L &= \frac{2W}{I^2}; \end{aligned}$$

Для подкласса «Энергия электрического поля»:

$$\begin{aligned} C &= \frac{q}{U}; & E_{\text{эл}} &= \frac{q^2}{2C}; \\ q &= CU; & E_{\text{эл}} &= \frac{CU^2}{2}. \\ U &= \frac{q}{C}; \end{aligned}$$

Задание 15

Проверьте работу модели, решив задачи, приведенные ниже. .
Список задач индивидуален для каждого варианта и определяется по табл. 6.4.11.

6. Лабораторные работы

Таблица 6.4.11. Распределение задач по вариантам

№ задачи	Вариант 1	Вариант 2	Вариант 3	Вариант 4	Вариант 5	Вариант 6	Вариант 7	Вариант 8	Вариант 9	Вариант 10	Вариант 11	Вариант 12	Вариант 13	Вариант 14	Вариант 15	Вариант 16	Вариант 17	Вариант 18	Вариант 19	Вариант 20
1	+			+		+		+	+		+		+	+		+		+	+	
2	+	+			+		+		+	+		+		+	+		+		+	+
3		+	+			+		+		+	+		+		+	+		+		+
4	+		+	+			+		+		+	+		+		+	+		+	
5		+		+	+			+		+		+	+		+		+	+		+
6	+		+		+	+			+		+		+	+		+		+	+	
7	+	+		+		+	+			+		+		+	+		+		+	+
8		+	+		+		+	+			+		+		+	+		+		+
9	+		+	+		+		+	+			+		+		+	+		+	
10		+		+	+		+		+	+			+		+		+	+		+
11	+		+		+	+		+		+	+			+		+		+	+	
12	+	+		+		+	+		+		+	+			+		+		+	+
13		+	+		+		+	+		+		+	+			+		+		+
14			+	+		+		+	+		+		+	+		+		+		+

Типовые задачи

Задача 1

Пешеход начинает движение от дома с координатой 50 м до автобусной остановки. Скорость пешехода 5,4 км/ч. В какой момент времени пешеход пройдет мимо остановки, если ее координата равна $x = 100$ м? ($t = \frac{x - x_0}{v_x}$) (33,3 с).

Задача 2

Расстояние от Москвы до Казани равно 810 км. С какой скоростью автобус преодолет заданное расстояние, если он затратит на дорогу 12 часов? ($v_x = \frac{S_x}{t}$) (18,75 м/с).

Задача 3

Спортсмен на последних метрах дистанции ускорился с 3 м/с до 5 м/с, преодолев этот участок за 100 с. Найдите ускорение спортсмена и дистанцию, которую он преодолел за это время.

$$(a = \frac{v - v_0}{\Delta t}; S_x = \frac{v^2 - v_0^2}{2a}) (0,02 \text{ м/с}^2; 400 \text{ м}).$$

Задача 4

Тело движется по окружности радиусом 50 м с линейной скоростью 10 м/с. Найдите центростремительное ускорение тела.

$$(a_n = \frac{v^2}{r}) (2 \text{ м/с}^2).$$

Задача 5

С каким ускорением двигался при разбеге реактивный самолет массой 60 т, если сила тяги двигателей 90 кН? ($a = \frac{F}{m}$) (1,5 м/с).

Задача 6

Шарик бросили вертикально вниз с высоты без начальной скорости. Он упал на землю через полминуты. Найдите высоту, с которой был брошен шарик, и скорость в момент падения на

землю. ($h = v_0 \cdot t + \frac{g t^2}{2}; v = v_0 + g t$) (4 410 м; 294 м/с).

6. Лабораторные работы

Задача 7

Найдите силу тяготения между телом массой 10 кг и телом массой 1 т, если расстояние между ними 10 м. ($F_{\text{тяг}} = G \frac{m_1 m_2}{r^2}$) ($6,67 \cdot 10^{-9}$ Н).

Задача 8

Рассчитайте массу Луны, если масса Земли равна $6 \cdot 10^{24}$ кг. Расстояние между Луной и Землей считать равным 384 000 км. Сила притяжения равна 1,9 · 10²⁰ Н. ($m_1 = \frac{F_{\text{тяг}} r^2}{G m_2}$) ($7 \cdot 10^{22}$).

Задача 9

Футбольному мячу массой 400 г при выполнении пенальти сообщили скорость 25 м/с. Какой при этом импульс придали мячу? ($p = mv$) (10 кг·м/с).

Задача 10

Импульс тела равен 8 кг·м/с, а его скорость 4 м/с. Найти массу тела. ($m = \frac{p}{v}$) (2 кг).

Задача 11

Найдите длину проводника, находящегося в магнитном поле с индукцией 10 Тл, при силе магнитного поля 0,07 кН и силе тока 0,5 А. ($l = \frac{F}{BI}$) (14 м).

Задача 12

Период вертикальных колебаний железнодорожного вагона равен 0,5 с. Определите частоту колебаний вагона. ($\nu = \frac{1}{T}$) (2 Гц).

6. Лабораторные работы

Задача 13

По обмотке соленоида индуктивностью 0,2 Гн течет ток 10 А. Определить энергию магнитного поля соленоида. ($W = \frac{LI^2}{2}$) (10 Дж).

Задача 14

Рассчитайте энергию электростатического поля конденсатора емкостью 0,1 мкФ, заряженного до разности потенциалов 200 В. ($E_{\text{эл}} = \frac{CU^2}{2}$) (2 мДж).

Задание 16

Сделать отчет по плану о выполнении лабораторной работы. План выполнения отчета находится в файле «Отчет по лабораторным работам».

Задание 17

Подготовить ответы на вопросы по теоретическому материалу. Защитить выполнение лабораторной работы:

- 1) показать работающую модель;
- 2) представить отчет в бумажном виде.

6.5. Лабораторная работа № 4

Модель «Генетика»

Описание

Данная модель позволяет определить наследуемые от родителей характеристики ребенка, такие как:

- группа крови;
- резус фактор;
- тип волос;

6. Лабораторные работы

- цвет волос;
- цвет глаз.

Необходимо создать модель, в которой будут использоваться классы, параметры, отношения и правила, описанные ниже.

Задание 1

Определите следующие классы:

- мама;
- папа;
- ребенок.

Задание 2

В каждый из созданных классов добавьте параметры из табл. 6.5.1.

Таблица 6.5.1. Параметры классов «Мама», «Папа», «Ребенок»

Обозначение	Тип параметра	Возможные исходные значения	Возможные выходные значения
Группа крови	Текст	«1», «2», «3», «4»	Возможные выходные значения «Все кроме первой», «Первая», «Первая либо вторая», «Первая либо третья», «Вторая либо третья», «Любая»
Резус-фактор	Текст	«Положительный», «Отрицательный»	«Положительный», «Отрицательный»
Тип волос	Текст	«Прямые», «Вьющиеся»	«Прямые», «Вьющиеся»
Цвет волос	Текст	«Темный», «Светлый», «Рыжий»	«Темный», «Светлый», «Рыжий», «Светлый либо рыжий»
Цвет глаз	Текст	«Карие», «Зеленые», «Каре-Зеленые», «Голубые», «Серые».	«Карие», «Зеленые», «Каре-Зеленые», «Голубые», «Серые», «Карие либо Зеленые», «Голубые либо серые»

Задание 3

Добавьте в модель отношения сложного типа со следующими названиями и содержанием:

1. Название: «Группа крови (отношение)»

Код отношения.

```
var m, f, r, b1="1", b2="2", b3="3", b4="4", allbut="все, кроме первой", b1l="первая",
b12l="первая либо вторая", b13l="первая либо третья", b23l="вторая либо третья",
any="любая";
m=String(m).toLowerCase();
f=String(f).toLowerCase();
if ((m==b1 && f==b4) || (f==b1 && m==b4)) {
    r=b23l;
}
else {
    if (m==b4 || f==b4) {
        r=allbut;
    }
    else {
        if (m==b1 && f==b1) {
            r=b1l;
        }
        else {
            if ((m==b2 && f==b3) || (f==b2 && m==b3)) {
                r=any;
            }
            else {
                if (m==b2 && f==b2) {
                    r=b12l;
                }
                else {
                    if (m==b3 && f==b3) {
                        r=b13l;
                    }
                    else {
                        r="Исходные данные
заданы неверно!";
                    }
                }
            }
        }
    }
}
}
```

В качестве входных параметров, добавьте следующие:

- *m*, типа «текст»;
- *f*, типа «текст».

6. Лабораторные работы

В качестве выходного параметра, добавьте следующий:

- *r*, типа «текст».

2. Название: «Цвет глаз(отношение)».

Код отношения.

```
var m, f, r="", brown="карие", green="зеленые", brgr="каре-зеленые", blue="голубые",
gray="серые", brorgr="карие либо зеленые", blorgr="голубые либо серые";
m=String(m).toLowerCase();
f=String(f).toLowerCase();
if (m==brgr || f==brgr) {
    r=brgr;
}
else {
    if ((m==brown && f==green) || (f==brown && m==green)) {
        r=brorgr;
    }
    else {
        if (m==brown || f==brown) {
            r=brown;
        }
        else {
            if (m==green || f==green) {
                r=green;
            }
            else {
                if ((m==blue && f==gray) || (f==blue
&& m==gray)) {
                    r=blorgr;
                }
                else {
                    if (m==blue || f==blue) {
                        r=blue;
                    }
                    else {
                        if (m==gray ||
f==gray) {
                            r=gray;
                        }
                        else {
                            r="Исходные данные заданы неверно!";
                        }
                    }
                }
            }
        }
    }
}
```

6. Лабораторные работы

В качестве входных параметров, добавьте следующие:

- m , типа «текст»;
- f , типа «текст».

В качестве выходного параметра, добавьте следующий:

- r , типа «текст».

3. Название: «Цвет волос (отношение)».

Код отношения

```
var m, f, r="", dark="темный", light="светлый", red="рыжий", rol="светлый либо
рыжий";
m=String(m.toLowerCase());
f=String(f.toLowerCase());
if (m==dark || f==dark) {
    r=dark;
}
else {
    if ((m==light && f==red) || (f==light && m==red)) {
        r=rol;
    }
    else {
        if (m==light && f==light) {
            r=light;
        }
        else {
            if (m==red && f==red) {
                r=red;
            }
            else {
                r="Исходные данные заданы неверно!";
            }
        }
    }
}
}
```

В качестве входных параметров, добавьте следующие:

- m , типа «текст»;
- f , типа «текст».

В качестве выходного параметра, добавьте следующий:

- r , типа «текст».

4. Название: «Тип волос (отношение)».

Код отношения.

```
var m, f, r, x="положительный", y="отрицательный";
m=String(m).toLowerCase();
f=String(f).toLowerCase();
if (m==x || f==x) {
    r=x;
}
else {
    if (m==y && f==y) {
        r=y;
    }
    else {
        r="Исходные данные заданы неверно!";
    }
}
}
```

В качестве входных параметров, добавьте следующие:

- *m*, типа «текст»;
- *f*, типа «текст».

В качестве выходного параметра, добавьте следующий:

r, типа «текст».

5. Название: «Резус-фактор (отношение)».

Код отношения.

```
var m, f, r, x="положительный", y="отрицательный";
m=String(m).toLowerCase();
f=String(f).toLowerCase();
if (m==x || f==x) {
    r=x;
}
else {
    if (m==y && f==y) {
        r=y;
    }
    else {
        r="Исходные данные заданы неверно!";
    }
}
}
```

6. Лабораторные работы

В качестве входных параметров, добавьте следующие:

- m , типа «текст»;
- f , типа «текст».

В качестве выходного параметра, добавьте следующий:

- r , типа «текст».

6. Название: «*Полное описание (отношение)*».

Код отношения.

```
var haircolor, hairtype, eyecolor, blood, rezus;  
y="цвет волос - " + haircolor + ", тип волос - "+ hairtype + ", цвет глаз - " +  
eyecolor + ", группа крови - " + blood + ", резус-фактор - " + rezus;
```

В качестве входных параметров, добавьте следующие:

- m , типа «текст»;
- f , типа «текст».

В качестве выходного параметра, добавьте следующий:

- r , типа «текст».

Задание 4

Для каждого отношения в модели добавьте соответствующее правило, подставляя соответствующие параметры из существующих классов.

Задание 5

Задавая различные значения параметров в классах «Мама» и «Папа» проверьте работоспособность модели. Допустимые значения указаны в табл. 1.

6.6. Лабораторная работа № 5

Описание предметной области

Аналитическое моделирование РСОД методом фонового потока

Формализованная схема и исходные данные РСОД

Общая формализованная схема распределенной системы массового обслуживания (РСОД) в виде сети массового обслуживания (СМО) приведена на рис. 6.6.1.

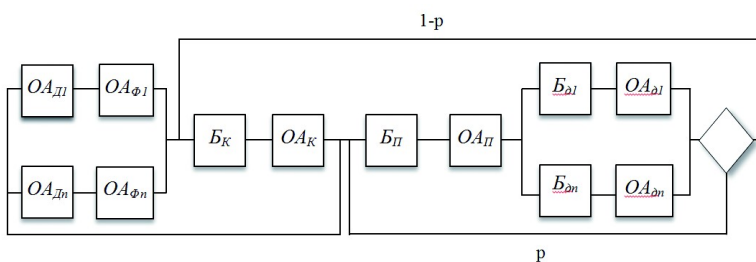


Рисунок 6.6.1. Формализованная схема РСОД, содержащая ПЭВМ, канал и сервер

$OA_{Дi}$ — обслуживающий аппарат, имитирующий дообработку на i -ной рабочей станции сети запроса от этой станции к серверу после обработки запроса на сервере;

$OA_{Фi}$ — обслуживающий аппарат, имитирующий формирование запроса от i -ной рабочей станции к серверу; ($i = \overline{1...N}$);

BK — буфер, имитирующий очередь запросов к каналу;

OAK — обслуживающий аппарат, имитирующий задержку при передаче данных через канал;

$BП$ — буфер, имитирующий очередь запросов к процессорам;

$OАП$ — обслуживающие аппараты, имитирующие работу процессоров;

$B_{дi}$ — буфер, имитирующий очередь запросов к i -му диску;

$OА_{дi}$ — обслуживающий аппарат, имитирующий работу i -го диска;

P — вероятность обращения запроса к ЦП после обработки на диске.

Обслуживание заявок во всех OA подчиняется экспоненциальному закону.

6. Лабораторные работы

Таблица 6.6.1. Исходные данные аналитической модели

Обозначение	Описание
N	Число рабочих станций сети
T_0	Среднее значение времени дообработки на рабочей станции сети запроса от этой станции к базе данных на сервере
T_p	Среднее значение времени формирования запроса от рабочей станции сети к базе данных на сервере
t_k	Среднее значение времени передачи запроса по каналу
C	Число процессоров сервера
t_{np}	Среднее значение времени обработки запроса в ЦП сервера
$t_{\Delta i}$	Среднее значение времени обработки запроса в диске сервера
P_i	Вероятность обращения запроса к i -му диску сервера после обработки запроса в процессоре
G	Вероятность обращения запроса к ЦП после обработки на диске

Таблица 6.6.2. Выходные характеристики аналитической модели

Обозначение	Описание
$T_{\text{реак}}$	Среднее значение времени реакции системы
ρ_k	Коэффициент загрузки OA , имитирующего работу канала передачи данных
ρ_{np}	Коэффициент загрузки OA , имитирующего работу процессора сервера
$\rho_{\Delta i}$	Коэффициент загрузки OA , имитирующего работу i -го диска сервера

Введем следующие обозначения:

$\lambda_{\phi 1}$ — среднее значение суммарной интенсивности фонового потока запросов, выходящих из OA , имитирующих работу рабочих станций, в канал;

$\lambda_{\phi 1\beta}$ — среднее значение интенсивности фонового потока запросов, проходящих через OA , имитирующих работу сервера и дисков, где $\beta = \frac{1}{1-p}$;

6. Лабораторные работы

β — среднее количество проходов запроса по тракту процессор-диски за время одного цикла его обработки в системе;

t_k — среднее значение времени обработки запроса в канале передачи данных; $t_k = 0,5(t_{k1} + t_{k2})$, где t_{k1} и t_{k2} , соответственно, среднее время передачи запроса по каналу в прямом и обратном направлениях;

n — количество серверов, обслуживающих рабочие станции;

$m = \frac{1}{P_i}$ — количество дисков в сервере, при условии, что все они одинаковые;

P_i — вероятность обращения к i -му диску сервера.

Порядок расчета рассматриваемой системы методом фонового потока

При расчете используется приближенный итерационный алгоритм нахождения значения выходных характеристик рассматриваемой системы.

1. Определяем начальное значение для $\lambda_{\phi 1}$:

$$\lambda_{\phi 1} = K_1 \cdot \min \left\{ \left[\frac{1}{2} t_k, \frac{C}{\beta t_{np}}, \frac{1}{\beta P_i t_d} \right] \frac{N-1}{N} \right\};$$

K_1 принимает значения в диапазоне 0,995...0,99995.

2. Определяем средние времена пребывания запроса в узлах системы: канале, процессоре, дисках:

$$T_k = \frac{2 t_k}{1 - 2 \lambda_{\phi 1} t_k};$$
$$T_{np} = \frac{\beta t_{np}}{1 - \left(\beta \lambda_{\phi 1} \frac{t_{np}}{c} \right)^c};$$
$$T_d = \frac{\beta t_d}{1 \beta P_i \lambda_{\phi 1} t_d}.$$

6. Лабораторные работы

3. Определяем интенсивность фонового потока после очередной итерации:

$$\lambda_{\phi} = \frac{N-1}{T_o + T_p + T_k + T_{np} + T_d}.$$

4. Сравниваем $\lambda_{\phi 1}$ и λ_{ϕ} . Если $\frac{|\lambda_{\phi 1} - \lambda_{\phi}|}{\lambda_{\phi}} < \Delta_1$, то переход на пункт 6, иначе на пункт 5.

5. Определяем новое приближённое значение для $\lambda_{\phi 1}$:

$$\delta_1 = \frac{\lambda_{\phi 1} - \lambda_{\phi}}{K_2}.$$

K_2 принимает значения в диапазоне 10...1000:

$$\lambda_{\phi 1} = \lambda_{\phi} - \delta_1.$$

Переход на пункт 2.

6. Определяем выходные результаты аналитической модели.

Определяем средние времена пребывания запроса в узлах системы: канале, процессоре и дисках:

$$T_k = \frac{2t_k}{1 - 2\lambda_{\phi 1}t_k};$$
$$T_{np} = \frac{\beta t_{np}}{1 - \left(\beta \lambda_{\phi 1} \frac{t_{np}}{c}\right)^c};$$
$$T_d = \frac{\beta t_d}{1 - \beta p_i \lambda_{\phi 1} t_d}.$$

Определяем загрузку основных узлов системы: рабочей станции, пользователя, канала передачи данных, процессора и дисков сервера.

$$\rho_{PC} = \frac{T_o + T_p}{T_{цикла}};$$
$$\rho_{польз} = \frac{T_p}{T_{цикла}};$$

6. Лабораторные работы

где $T_{\text{цикла}} = T_o + T_p + T_k + T_{\text{пр}} + T_d$;

$$\rho_k = 2\lambda t_k;$$

$$\rho_{\text{пр}} = \beta \lambda \frac{t_{\text{пр}}}{c};$$

$$\rho_d = \beta \lambda p_i t_d;$$

$$\text{где } \lambda = \frac{N}{T_{\text{цикла}}}.$$

Задание 1. Создание классов и параметров

1.1. Создайте три класса:

- исходные данные;
- выходные данные;
- фоновый поток.

В классе «Фоновый поток» создайте подкласс «Работа в цикле».

1.2. Добавление параметров.

В класс «Исходные данные» добавьте следующие параметры из табл. 6.6.3 и опишите их, при необходимости значения параметров задайте по умолчанию.

Таблица 6.6.3. Исходные данные

Обозначение	Тип параметра	Описание
C	Число	Число процессоров сервера
Δ	Число	Точность расчетов
G	Число	Вероятность обращения запроса к ЦП после обработки на диске
K_1	Число	Принимает значения в диапазоне [0,995...0,99995]
K_2	Число	Принимает значения в диапазоне 10...1000
M	Число	Количество дисков в сервере
N	Число	Число рабочих станций сети
T_o	Число	Среднее значение времени дообработки на рабочей станции сети запроса от этой станции к базе данных на сервере
T_p	Число	Среднее значение времени формирования запроса от рабочей станции сети к базе данных

6. Лабораторные работы

T_{cpu}	Число	Среднее значение времени обработки запроса в центральном процессоре сервера
T_{di}	Число	Среднее значение времени обработки запроса в диске сервера
t_{k1}	Число	Среднее значение времени передачи запроса по каналу в прямом направлении
t_{k2}	Число	Среднее значение времени передачи запроса по каналу в обратном направлении

В класс «Выходные данные» добавьте параметры из табл. 6.6.4 и опишите их.

Таблица 6.6.4. Выходные данные

Обозначение	Тип параметра	Описание
$PoCPU$	Число	Коэффициент загрузки OA , имитирующего работу процессора сервера
$PoDi$	Число	Коэффициент загрузки OA , имитирующего работу i -го диска сервера
PoK	Число	Коэффициент загрузки OA , имитирующего работу канала передачи данных
P_{pc}	Число	Коэффициент загрузки рабочей станции
P_u	Число	Коэффициент загрузки пользователя рабочей станции
T_{loop}	Число	Время цикла системы
T_r	Число	Время реакции системы

В класс «Фоновый поток» добавьте параметры из табл. 6.6.5 и опишите их.

Таблица 6.6.5. Фоновый поток

Обозначение	Тип параметра	Описание
β	Число	Среднее количество проходов запроса по тракту процессор-диски за время одного цикла его обработки в системе
Min	Число	Минимальное число
P_i	Число	Вероятность обращения к i -му диску сервера

6. Лабораторные работы

t_k	Число	Среднее значение времени обработки запроса в канале передачи данных
-------	-------	---

Класс «Работа в цикле» является вложенным классом в «Фоновый поток». Добавьте в класс «Работа в цикле» параметры из табл. 6.6.6 и опишите их.

Таблица 6.6.6. Работа в цикле

Обозначение	Тип параметра	Описание
I	Число	Число итераций в цикле
$T_{срi}$	Число	Среднее время обработки запроса процессором
T	Число	Среднее время пребывания запроса в канале
T_d	Число	Средние времена пребывания запроса в дисках

В результате у вас должны быть сформированы классы с параметрами (рис. 6.6.2).



Рисунок 6.6.2. Сформированные в модели классы и их параметры

6. Лабораторные работы

Задание 2. Создание отношений и добавление правил

Создайте отношения по табл. 6.6.7, которые будут использоваться в модели, и опишите их.

Таблица 6.6.7. Отношения в модели

Формула	Входные параметры	Выходные параметры	Тип
$a=(b+c)/2$	b, c	A	Формула
$a \geq 10$	a	–	Ограничение
$a \leq 1000$	–	–	Ограничение
$a \geq 0,995$	–	–	Ограничение
$a \leq 0,99995$	–	–	Ограничение
$a \geq 0$	–	–	Ограничение
$a < 1$	–	–	Ограничение
$y1=x1+x2+x3+x4+x5;$ $z=z1/y1;$ $y2=(x1+x2)/y1;$ $y3=x2/y1;$ $y4=2 \cdot z \cdot x6;$ $y5=b \cdot z \cdot x7/x8;$ $y6=b \cdot z \cdot x9 \cdot x10$	$x1, x2, x3, x4, x5,$ $x6, x7, x8, x9,$ $x10, z, z1, b$	$y1, y2, y3, y4, y5,$ $y6$	Сложное отношение
$y=x1+x2+x3+x4$	$x1, x2, x3, x4$	Y	Формула
$y=1/(1-x)$	x	Y	Формула
$y=1/x$	x	Y	Формула
<pre> if (tk==0) { a=999999 } else a=1/(2*tk); b=CL/(bL*tp); c=1/(bL*Pi*td); if (a<b) { if (a<c) {d=a;} else d=c; } else { if (b<c) { d=b;} else d=c; } elsed=b;</pre>	tk, CL, bL, Pi, td, tp	D	Сложное отношение

6. Лабораторные работы

Формула	Входные параметры	Выходные параметры	Тип
<pre> Lf2=K1*min*(N-1)/N; temp=2; i=0; d=i; while (temp>delta) { d=i+1; i=d; Tk=2*tk/(1-2*Lf2*tk); Tcpu=b*tcpu/(1- Math.pow(b*Lf2*tcpu/C,C)); Td=b*tdi/(1-b*Pi*Lf2*tdi); Lf=(N-1)/ (T0+Tp+Tk+Tcpu+Td); sigma=(Lf2-Lf)/K2; Lf1=Lf2; Lf2=Lf1-sigma; temp=Math.abs((Lf1- Lf)/Lf); } Tk=2*tk/(1-2*Lf1*tk); Tcpu=b*tcpu/(1- Math.pow(b*Lf1*tcpu/C,C)); Td=b*tdi/(1-b*Pi*Lf1*tdi); </pre>	<p>K1, K2, min,N, delta, tk, b, tcpu, C, tdi, Pi, T0, Tp</p>	<p>Tk, Td, Tcpu, i</p>	Сложное отношение

Добавьте правила, реализующие следующие формулы:

$$\begin{aligned}
 b &= \frac{1}{1-p}; & P_{pc} &= \frac{T_0 + T_p}{T_{loop}}; \\
 t_k &= 0,5(t_{k1} + t_{k2}); & P_u &= \frac{T_p}{T_{loop}}; \\
 P_i &= \frac{1}{M}; & PoK &= 2l t_k; \\
 min &= min \left\{ \frac{1}{2t_k}; \frac{C}{b t_{cpu}}; \frac{1}{b P_i t_{di}} \right\} & PoCPU &= b l \frac{t_{cpu}}{C}; \\
 & & PoDi &= b l P_i t_{di}; \\
 T_{loop} &= T_0 + T_p + T_k + T_{cpu} + T_d; & T_r &= T_p + T_k + T_{cpu} + T_d. \\
 l &= \frac{N}{T_{loop}};
 \end{aligned}$$

Также добавьте правила (см. соответствующие отношения в табл. 6.6.7), вычисляющие минимальное значение по формуле:

6. Лабораторные работы

$$y = \min \left\{ \frac{1}{2 \cdot x_1}; \frac{x_2}{x_3 \cdot x_4}; \frac{1}{x_3 \cdot x_5 \cdot x_6} \right\}.$$

Также введите правила, реализующие цикл системы (отношение №13).

Добавьте ограничения на K_1 , K_2 и G .

В результате у вас должны быть сформированы отношения, правила и ограничения (рис. 6.6.3).

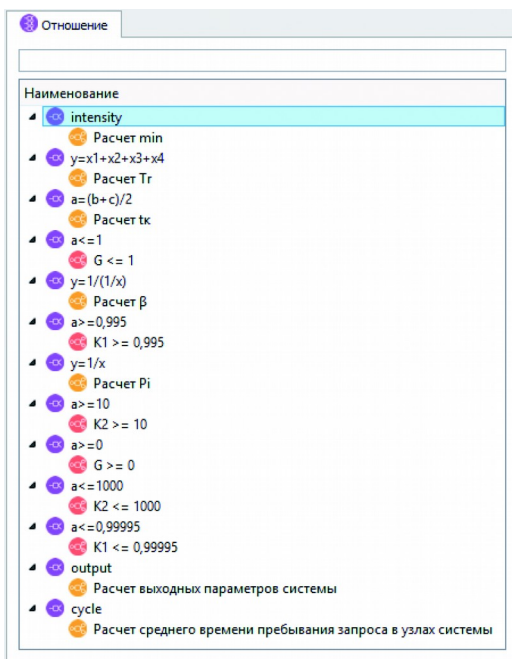


Рисунок 6.6.3. Отношения, правила и ограничения в модели

Задание 3. Проверка работы модели

Выполните тестирование модели.

1. Введите в поле K_2 значение, не попадающее в интервал 10...100, выберите одно или несколько выходных значений,

6. Лабораторные работы

нажмите кнопку «Запустить». В результате в консоли должно появиться сообщение «Не выполнено ограничение» (рис. 6.6.4).

The screenshot shows a test environment window titled "Тест: Лабораторная работа №5". It contains a table with columns "Объект", "Значение", and "Найти". The table is organized into two main sections: "Выходные данные" and "Исходные данные".

Объект	Значение	Найти
▲ Лабораторная работа №5		
▲ Выходные данные		
РoCPU	Число	<input checked="" type="checkbox"/>
РoDi	Число	<input type="checkbox"/>
РoK	Число	<input type="checkbox"/>
Ррс	Число	<input type="checkbox"/>
Ри	Число	<input type="checkbox"/>
Тлоор	Число	<input type="checkbox"/>
Тг	Число	<input type="checkbox"/>
▲ Исходные данные		
С	Число	<input type="checkbox"/>
G	Число	<input type="checkbox"/>
K1	Число	<input type="checkbox"/>
K2	0	<input type="checkbox"/>
M	Число	<input type="checkbox"/>
N	Число	<input type="checkbox"/>
t_cpu	Число	<input type="checkbox"/>
t_di	Число	<input type="checkbox"/>
t_k1	Число	<input type="checkbox"/>

Below the table, there are tabs for "Ошибки" (Errors) and "Консоль" (Console). The console output shows the following message:

```
Не выполнено ограничение:  
382fa810-607b-4a97-b231-ed5c2cc98a31 | if (a >= 10) {flag = true;} else {flag = false;} |  
K2()=0
```

Рисунок 6.3. Проверка на выполнение ограничений

2. Продемонстрируйте выполнение других ограничений. Заполните табл. 6.6.8.

6. Лабораторные работы

Таблица 6.6.8. Результаты моделирования

Эксперимент	1	2	3	4	5
<i>Исходные данные</i>					
Количество рабочих станций (N)	10	10	20	20	30
Среднее время дообработки запроса на РС (T_0)	50	100	50	100	50
Среднее время формирования запроса на РС (T_p)	50	100	50	100	100
Среднее время передачи через канал в прямом направлении (t_{k1})	5	10	10	5	5
Среднее время передачи через канал в обратном направлении (t_{k2})	5	10	10	5	5
Количество процессоров (C)	1	2	1	2	1
Среднее время обработки запроса на процессоре (t_{cpu})	10	20	10	20	10
Количество дисков (m)	2	4	2	2	2
Среднее время обработки запроса на диске (t_d)	20	10	20	20	40
Вероятность обращения запроса к ЦП после обработки на диске (G)	0	0	0	0	0
<i>Результаты моделирования</i>					
Загрузка рабочей станции (P_{pc})					
Загрузка пользователя рабочей станции (P_u)					
Загрузка канала (P_{oK})					
Загрузка процессора (P_{oCPU})					
Загрузка диска (P_{oDi})					
Среднее время цикла системы (T_{loop})					
Среднее время реакции системы (T_r)					

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. История развития систем искусственного интеллекта.
2. Основные направления развития систем искусственного интеллекта.
3. Общая характеристика процессов сбора, передачи, хранения, обработки и представления информации.
4. Предметная область. Данные и знания. Свойства, характеристики знаний.
5. Миварные базы данных и искусственный интеллект.
6. Модель процессов передачи и хранения информации.
7. Три уровня исследований в области искусственного интеллекта.
8. Основные подходы и направления развития искусственного интеллекта.
9. Прикладной подход к развитию искусственного интеллекта.
10. Фундаментальный подход к развитию искусственного интеллекта.
11. Миварный подход к созданию искусственного интеллекта.
12. Миварная логико-вычислительная обработка.
13. Введение в представление знаний. Обработка знаний, выраженных в качественной форме.
14. Продукционные системы.
15. Представление данных в миварной эволюционной модели данных.
16. Операции над данными в миварных моделях данных.

17. Представление в теории миваров правил на основе продукций. Подходы к логической обработке данных и выводу.
18. Методы прямой и обратной волны логического вывода.
19. Представление продукций и сетей продукций в виде двудольных графов.
20. Развитие миварных сетей для реализации правил выбора «если..., то..., иначе...».
21. Области применения экспертных систем. Характеристики ЭС.
22. Разработка экспертных систем. Этапы разработки экспертных систем. Организация знаний в базе данных.

СПИСОК СОКРАЩЕНИЙ

- ИИ — искусственный интеллект.
ЛРС — логически рассуждающие системы.
СППР — системы поддержки принятия решений.
АСОИ — автоматизированные системы обработки информации.
СПАКОД — самоорганизующиеся программно-аппаратные комплексы оперативной диагностики.
АСУ — автоматизированные системы управления.
ЛИИ — логический искусственный интеллект.
ИППП — интеллектуальные пакеты прикладных программ.
ЭС — экспертные системы.
БЗ — база знаний.
MIVAR — Multidimensional Informational Variable Adaptive Reality.
МИВАР — Многомерная Информационная Варьирующаяся Адаптивная Реальность.
ГПДС — глобальные познающе-диагностические системы.
ЯОЗ — языки описания знаний.

СПИСОК ЛИТЕРАТУРЫ

Основная литература

1. Варламов О.О. Основы миварного подхода к созданию логического искусственного интеллекта: учеб. пособ. — М.: МАДИ, 2013. — 80 с.
2. Варламов О.О. Прикладная математика: гносеологические основы миварных технологий создания систем искусственного интеллекта: учеб. пособ. — М.: МАДИ, 2013. — 84 с.
3. Варламов О.О. Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство. — М.: Радио и связь, 2002. — 288 с.
4. Варламов О.О., Чибирова М.О., Хадиев А.М., Антонов П.Д., Сергушин Г.С., Протопопова Д.А., Жданович Е.А., Збавитель П.Ю., Сараев Д.В., Шошев И.А., Петерсон А.О. Практикум по миварному моделированию и созданию экспертных систем (на примере программного комплекса «Конструктор экспертных систем МИВАР 1.1» (КЭСМИ 1.1). учеб. пособ. / под ред. О.О. Варламова. — М.: НИИ МИВАР, 2015. — 246 с.
5. Васильев В.И., Ильясов Б.Г. Интеллектуальные системы управления. Теория и практика: учеб. пособ. — М.: Радиотехника, 2009. — 392 с.
6. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. — СПб.: Питер, 2001. — 384 с.
7. Гладков Л.А., Курейчик В.В., Курейчик В.М. Дискретная математика. — М.: ФИЗМАТЛИТ, 2014. — 496 с.
8. Джерратано Д., Райли Г. Экспертные системы: принципы разработки и программирование. 4-е изд. — М.: Вильямс, 2007. — 1152 с.

9. Кузнецов О.П. Дискретная математика для инженера. — СПб.: Лань, 2009. — 400 с.
10. Люгер Дж.Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. 4-е изд. — М.: Вильямс, 2005. — 864 с.
11. Осипов Г.С. Методы искусственного интеллекта. — М.: Физматлит, 2011. — 296 с.
12. Поспелов Д.А. Моделирование рассуждений. Опыт анализа мыслительных актов. — М.: Радио и связь, 1989. — 184 с.
13. Райбекас А.Я. Вещь, свойство, отношение как философские категории. — Томск: Изд-во Томского университета, 1977. — 245 с.
14. Рассел С., Норвиг П. Искусственный интеллект: современный подход. 2-е изд. — М.: Вильямс, 2006. — 1408 с.
15. Ростовцев Ю.Г. Математические методы и модели оценивания военно-политической обстановки. — М.: МО СССР, 1986. — 312 с.
16. Buchanan B.G., Smith R.G. Fundamentals of Expert Systems // Annual Review of Computer Science, 1988, vol. 3, pp. 23–57.
17. Frege G. Die Grundlagen der Arithmetik. — Breslau: W. Koeber, 1984. 119 p.
18. Newell A., Simon H. Human Problem Solving. Englewood Cliffs. — NJ: Prentice Hall, 1972. 920 p.
19. Tarski A. Logic, semantics, metamathematics: papers from 1923 to 1938. — Clarendon Press, 1956. 471 p.
20. Whitehead A.N., Russel B. Principia Mathematica. 2nd ed. — Cambridge: Cambridge University Press, 1925–1927. 1908 p.
21. Winograd T., Flores F. Understanding computers and cognition. — Norwood, N.J.: Ablex, 1986. 207 p.

Дополнительная литература

22. Адамова Л.Е., Белоусова А.И., Протопопова Д.А., Елисеев Д.В., Петерсон А.О. Об одном подходе к созданию интеллектуальной вопросно-ответной системы «Миварный виртуальный консультант» // Радиопромышленность. 2015. № 3. С. 160–171.
23. Адамова Л.Е., Петерсон А.О., Протопопова Д.А., Скакунова Е.А., Варламов О.О. Исследование подходов и основных проблем понимания естественного русского языка // Автоматизация и управление в технических системах. 2014. № 2. С. 107–122.
24. Адамова Л.Е., Скакунова Е.А., Варламов О.О., Петерсон А.О., Протопопова Д.А. Миварное понимание текста: разработка методики обучения виртуальной личности предметным знаниям на основе создания сетей концептов // Автоматизация и управление в технических системах. 2014. № 3. С. 48–65.
25. Антонов П.Д., Чибирова М.О., Жданович Е.А., Сергушин Г.С., Елисеев Д.В. Практический пример использования миварного подхода для создания экспертной системы в предметной области «геометрия» // Радиопромышленность. 2015. № 3. С. 131–143.
26. Белоусова А.И., Алексахин С.В., Варламов О.О., Строганов Д.В., Суркова Н.Е. Применение мультиагентной системы при дистанционном обучении персонала промышленных предприятий // В мире научных открытий. 2012. № 2.6. С. 175–183.
27. Белоусова А.И., Варламов О.О. Использование миваров и многоуровневой модели гетерогенной мультиагентной системы на практике // Известия Кабардино-Балкарского научного центра РАН. 2011. № 1. С. 39–45.

28. Варламов О.О. Алгоритм разреза сети по вершинам и ребрам ее графа сложности $O(n^2)$ // Труды НИИР: Сб. ст. 1997. С. 92–97.
29. Варламов О.О. Анализ взаимосвязей GRID и САС ИВК, SOA и миварного подхода // Искусственный интеллект. 2005. № 4. С. 4–11.
30. Варламов О.О. Интеллектуальные системы информационной безопасности и системный синтез модели компьютерных угроз // Искусственный интеллект. 2006. № 3. С. 720–727.
31. Варламов О.О. Миварные технологии: переход от продукций к двудольным миварным сетям и практическая реализация автоматического конструктора алгоритмов, управляемого потоком входных данных и обрабатывающего более трех миллионов продукционных правил // Искусственный интеллект. 2012. № 4. С. 11–33.
32. Варламов О.О. Миварный подход к разработке интеллектуальных систем и проект создания мультипредметной активной миварной интернет-энциклопедии // Известия Кабардино-Балкарского научного центра РАН. 2011. № 1. С. 55–64.
33. Варламов О.О. О возможности создания интеллектуальных систем на основе GRID, систем адаптивного синтеза ИВК, сервисно-ориентированной архитектуры и миварного информационного пространства // Известия Таганрогского государственного радиотехнического университета. 2005. Т. 54. № 10. С. 130–140.
34. Варламов О.О. О необходимости перехода от теории искусственного интеллекта к разработке теории активного отражения // Известия Южного федерального университета. Технические науки. 2007. Т. 77. № 2. С. 89–95.

35. Варламов О.О. О системном подходе к созданию модели компьютерных угроз и ее роли в обеспечении безопасности информации в ключевых системах информационной инфраструктуры // Известия ТРТУ. 2006. Т. 62. № 7. С. 216–223.
36. Варламов О.О. Обзор двадцати пяти лет развития миварного подхода к разработке интеллектуальных систем и создания искусственного интеллекта // Труды НИИР. 2011. № 1. С. 34–44.
37. Варламов О.О. Основы многомерного информационного развивающегося (миварного) пространства представления данных и правил // Информационные технологии. 2003. № 5. С. 42–47.
38. Варламов О.О. Параллельная обработка потоков информации на основе виртуальных потоковых баз данных // Известия высших учебных заведений. Электроника. 2003. № 5. С. 82–89.
39. Варламов О.О. Переборное единично-инкрементное суммирование чисел с линейной вычислительной сложностью // Автоматизация и современные технологии. 2003. № 1. С. 34–40.
40. Варламов О.О. Перспективы создания миварных систем управления для автономных интеллектуальных роботов // Радиопромышленность. 2015. № 3. С. 210–225.
41. Варламов О.О. Практическая реализация линейной вычислительной сложности логического вывода на правилах «если-то» в миварных сетях и обработка более трех миллионов правил // Автоматизация и управление в технических системах. 2013. № 1. С. 60–97.
42. Варламов О.О. Разработка адаптивного механизма логического вывода на эволюционной интерактивной сети гиперправил с мультиактивизаторами, управляемой потоком данных // Искусственный интеллект. 2002. № 3. С. 363–370.

43. Варламов О.О. Разработка квадратичной сложности методов поиска минимального разреза двухполюсных и многополюсных сетей // Искусственный интеллект. 2002. № 3. С. 371–375.
44. Варламов О.О. Разработка линейного матричного метода определения маршрута логического вывода на адаптивной сети правил // Известия вузов. Сер. «Электроника». 2002. № 6. С. 43–51.
45. Варламов О.О. Роль и место миваров в компьютерных науках, системах искусственного интеллекта и информатике // Радиопромышленность. 2015. № 3. С. 10–27.
46. Варламов О.О. Системный анализ и синтез моделей данных и методы обработки информации для создания самоорганизующихся комплексов оперативной диагностики // Искусственный интеллект. 2003. № 3. С. 299–305.
47. Варламов О.О. Системы обработки информации и взаимодействие групп мобильных роботов на основе миварного информационного пространства // Искусственный интеллект. 2004. № 4. С. 695–700.
48. Варламов О.О. Создание интеллектуальных систем на основе взаимодействия миварного информационного пространства и сервисно-ориентированной архитектуры // Искусственный интеллект. 2005. № 3. С. 13–17.
49. Варламов О.О. Создание теории активного отражения как обобщения теории искусственного интеллекта и возможность ее реализации в миварном инфопространстве // Искусственный интеллект. 2007. № 3. С. 17–24.
50. Варламов О.О. Формализация термина «понимание смысла текста» на основе миварных технологий и концепции «Вещь-Свойство-Отношение» // Радиопромышленность. 2015. № 3. С. 144–159.

51. Варламов О.О. Эволюционные базы данных и знаний. Миварное информационное пространство // Известия Южного федерального университета. Технические науки. 2007. Т. 77. № 2. С. 77–81.
52. Варламов О.О., Адамова Л.Е., Елисеев Д.В., Майборода Ю.И., Антонов П.Д., Сергушин Г.С., Чибирова М.О. Комплексное моделирование процессов понимания компьютерами смысла текстов, речи и образов на основе миварных технологий // Искусственный интеллект. 2013. № 4. С. 15–27.
53. Варламов О.О., Адамова Л.Е., Елисеев Д.В., Майборода Ю.И., Антонов П.Д., Сергушин Г.С., Чибирова М.О. О миварном подходе к моделированию процессов понимания компьютерами смысла текстов, речи и образов. Новые возможности расширения границ автоматизации умственной деятельности человека // Автоматизация и управление в технических системах. 2013. № 2. С. 38–51.
54. Варламов О.О., Адамова Л.Е., Петерсон А.О., Протопопова Д.А., Скакунова Е.А. Исследование подходов и основных проблем понимания естественного русского языка // Автоматизация и управление в технических системах. 2014. № 2. URL: auts.esrae.ru/10-196.
55. Варламов О.О., Антонов П.Д., Чибирова М.О., Хадиев А.М., Сергушин Г.С. Мивар: машино-реализуемый способ автоматизированного построения маршрута логического вывода в базе знаний // Радиопромышленность. 2015. № 3. С. 28–43.
56. Варламов О.О., Владимиров А.Н., Носов А.В., Потапова Т.С. Применение многопроцессорного вычислительного кластера НИИР для распараллеливания алгоритмов в научно-технических и вычислительных задачах // Труды Научно-исследовательского института радио. 2009. № 3. С. 120–123.

57. Варламов О.О., Владимиров А.Н., Носов А.В., Потапова Т.С. Программный комплекс «УДАВ»: практическая реализация активного обучаемого логического вывода с линейной вычислительной сложностью на основе миварной сети правил // Труды Научно-исследовательского института радио. 2010. Т. 1. С. 108–116.
58. Варламов О.О., Лазарев В.М., Атакищев О.И., Чибирова М.О., Сергушин Г.С. Анализ возможности использования когнитивных карт и миварных сетей для построения экспертных систем обработки данных космических систем наблюдения // Известия Юго-Западного государственного университета. Сер. «Управление, вычислительная техника, информатика. Медицинское приборостроение». 2014. № 1. С. 76–92.
59. Варламов О.О., Лазарев В.М., Осипов В.Г. О перспективах создания автономных интеллектуальных роботов на основе миварных технологий накопления и логической обработки информации // Известия ЮФУ. Технические науки. 2016. № 2. С. 22–38.
60. Варламов О.О., Майборода Ю.И., Сергушин Г.С., Хадиев А.М. Применение миварных экспертных систем для решения задач понимания текста и распознавания изображений // В мире научных открытий. 2015. № 6. С. 205–214.
61. Варламов О.О., Носов А.В., Владимиров А.Н., Потапова Т.С., Колупаева Е.Г. Программа «УДАВ»: реализация линейной вычислительной сложности матричного метода поиска маршрута логического вывода на основе миварной сети правил // Искусственный интеллект. 2009. № 3. С. 443–448.
62. Варламов О.О., Санду Р.А., Владимиров А.Н., Бадалов А.Ю. О создании на основе миварных сетей МЭПАИС — многомерной эволюционной прикладной автоматизированной информационной системы поддержки

принятия решений для управления инновационными ресурсами химической и нефтехимической промышленности России // Труды Научно-исследовательского института радио. 2010. № 3. С. 27–34.

63. Варламов О.О., Санду Р.А., Владимиров А.Н., Бадалов А.Ю., Тожа К.Э. Миварный метод логико-вычислительной обработки информации для АСУ, тренажеров и экспертных систем реального времени // Искусственный интеллект. 2010. № 4. С. 558–565.

64. Варламов О.О., Санду Р.А., Владимиров А.Н., Носов А.В., Оверчук М.Л. Миварный подход к созданию мультипредметных активных экспертных систем в целях обучения информационной безопасности и управления инновационными ресурсами в образовании // Известия Южного федерального университета. Технические науки. 2010. Т. 112. № 11. С. 226–232.

65. Варламов О.О., Тожа К.Э. Интеллектуальные системы принятия решений и запатентованное устройство определения в сети связи места минимального разреза и максимального потока // Искусственный интеллект. 2008. № 4. С. 302–307.

66. Варламов О.О., Чибирова М.О., Сергушин Г.С. Практическая реализация универсального решателя задач «УДАВ» с линейной сложностью логического вывода на основе миварного подхода и «облачных» технологий // Приборы и системы. Управление, контроль, диагностика. 2013. № 11. С. 45–55.

67. Варламов О.О., Чибирова М.О., Сергушин Г.С., Елисеев Д.В. «Облачная» реализация миварного универсального решателя задач на основе адаптивного активного логического вывода с линейной сложностью относительно правил «Если-То-Иначе» // Автоматизация и управление в технических системах. 2013. № 2. С. 7–23. URL: auts.esrae.ru/4-78.

68. Васюгова С.А., Варламов О.О. О возможностях использования миварных технологий представления знаний и обработки данных для групп роботов и гетерогенных мультиагентных систем и сред // Известия Кабардино-Балкарского научного центра РАН. 2011. № 1. С. 65–70.
69. Васюгова С.А., Остроух А.В., Краснянский М.Н., Самаратунга А. Исследование перспектив и проблем интеграции человека с компьютером: искусственный интеллект, робототехника, технологическая сингулярность и виртуальная реальность // Перспективы науки. 2011. № 4. С. 109–115.
70. Владимиров А.Н., Варламов О.О., Носов А.В., Потапова Т.С. Программный комплекс «УДАВ»: практическая реализация активного обучаемого логического вывода с линейной вычислительной сложностью на основе миварной сети правил // Труды НИИР. 2010. Т. 1. С. 108–116.
71. Владимиров А.Н., Варламов О.О., Санду Р.А., Бадалов А.Ю., Тожа К.Э. Теоретическое развитие миварных сетей для реализации правил выбора «ЕСЛИ.., ТО.., ИНАЧЕ..» на основе многодольных графов // Искусственный интеллект. 2011. № 1. С. 34–41.
72. Давыдова Т.Л., Варламов О.О., Остроух А.В., Краснянский М.Н. Анализ возможностей миварного подхода для систем искусственного интеллекта и современной робототехники // Вестник ТГТУ. Т. 17. № 3. С. 687–694.
73. Жданович Е.А., Панферов А.А., Юфимычев К.А., Хадиев А.М., Елисеев Д.В. Применение миварной экспертной системы для планирования движения мобильного сервисного робота // Радиопромышленность. 2015. № 3. С. 243–254.
74. Жданович Е.А., Антонов П.А., Хадиев А.М., Сергушин Г.С., Чибирова М.О. Постановка диагноза по симптомам на основе миварного подхода // Радиопромышленность. 2015. № 3. С. 122–130.

75. Жданович Е.А., Чернышев П.К., Юфимычев К.А., Елисеев Д.В., Чувииков Д.А. Вычисление произвольных алгоритмов функционирования сервисных роботов на основе миварного подхода // Радиопромышленность. 2015. № 3. С. 226–242.
76. Майборода Ю.И., Синцов М.Ю., Озерин А.Ю., Кузин А.А., Варламов О.О. Система автоматического тегирования изображений на основе миварных технологий // Программные системы: теория и приложения. 2014. № 4. С. 159–179.
77. Максимова А.Ю., Варламов О.О. Миварная экспертная система для распознавания образов на основе нечеткой классификации и моделирования различных предметных областей с автоматизированным расширением контекста // Известия Южного федерального университета. Технические науки. 2011. Т. 125. № 12. С. 77–87.
78. Медведев В.Е., Соломатин А.В., Варламов О.О., Жанказиев С.В., Ивахненко А.М. Решение задачи регулирования дорожного движения на основе автоматизированной системы управления // В мире научных открытий. 2012. № 2.6. С. 124–129.
79. Остроух А.В., Подкосова Я.Г., Варламов О.О., Краснянский М.Н. Анализ перспектив использования технологий виртуальной реальности в дистанционном обучении // Вопросы современной науки и практики. Университет им. В.И. Вернадского. 2011. № 2. С. 104–111.
80. Петерсон А.О. Разработка методики автоматизированного обучения миварной базы знаний в целях понимания компьютерами естественного русского языка // Автоматизация и управление в технических системах. 2014. № 4. URL: auts.esrae.ru/12-235.

81. Подкосова Я.Г., Васюгова С.А., Варламов О.О. Использование технологий виртуальной реальности для трехмерной визуализации результатов моделирования и для миварных обучающих систем // Известия Кабардино-Балкарского научного центра РАН. 2011. № 1. С. 226–232.
82. Подкосова Я.Г., Васюгова С.А., Варламов О.О. Новые возможности и ограничения технологий виртуальной реальности для проведения научных исследований, трехмерной визуализации результатов моделирования и создания миварных обучающих систем и тренажеров // Труды Научно-исследовательского института радио. 2011. № 2. С. 5–16.
83. Протопопова Д.А. Разработка методики формирования комплексной сети концептов в миварной базе знаний в целях решения задачи понимания текстов на естественном русском языке // Автоматизация и управление в технических системах. 2014. № 4. URL: auts.esrae.ru/12-234.
84. Протопопова Д.А., Варламов О.О. Использование миварного подхода для понимания текста, написанного на естественном языке // Автоматизация и управление в технических системах. 2015. № 4.1. URL: auts.esrae.ru/16-344.
85. Решетников Р.Б., Петерсон А.О., Протопопова Д.А., Хадиев А.М. Применение миварного подхода в перспективных АСУ в области добычи и транспортировки полезных ископаемых // Радиопромышленность. 2015. № 3. С. 112–121.
86. Санду Р.А., Варламов О.О. Миварный подход к созданию интеллектуальных систем и искусственного интеллекта. Результаты 25 лет развития и ближайшие перспективы. — М.: Стандартинформ, 2010. — 339 с.
87. Санду Р.А., Варламов О.О., Остроух А.В. Миварные автоматизированные системы управления технологическими процессами для нефтяной промышленности России //

Список литературы

Автоматизация, телемеханизация и связь в нефтяной промышленности. 2011. № 11. С. 37–40.

88. Сергушин Г.С. Компьютерно-реализованная система для автоматизированного построения маршрута логического вывода в миварной базе знаний // Радиопромышленность. 2015. № 3. С. 90–99.

89. Сергушин Г.С. Разработка миварных АСУ ТП для различных применений в автомобильно-дорожной сфере // Радиопромышленность. 2015. № 3. С. 100–111.

90. Сергушин Г.С., Варламов О.О., Чибирова М.О., Елисеев Д.В., Муравьева Е.А. Исследование возможностей информационного моделирования сложных систем управления технологическими процессами на основе миварных технологий // Автоматизация и управление в технических системах. 2013. № 2. С. 51–66.

91. Сергушин Г.С., Варламов О.О., Чибирова М.О., Елисеев Д.В., Муравьева Е.А. Информационное моделирование сложных автоматизированных систем управления технологическими процессами на основе миварных технологий // Искусственный интеллект. 2013. № 3. С. 126–138.

92. Сергушин Г.С., Протопопова Д.А. О преимуществах применения миварных технологий для автоматизации производственных процессов // Автоматизация и управление в технических системах. 2015. № 4.1. URL: auts.esrae.ru/16-345.

93. Синцов М.Ю., Озерин А.Ю., Кузин А.А., Варламов О.О. О развитии миварного подхода к интеллектуальному распознаванию образов для работы с трехмерными объектами // Радиопромышленность. 2015. № 3. С. 172–183.

94. Скворцова А.Ю., Варламов О.О., Сергушин Г.С., Белоусова А.И. Исследование возможностей практического применения технологий виртуальной реальности и угрозы ее развития // Автоматизация и управление в технических системах. 2014. № 2. С. 98–106.

95. Соломатин А.В., Варламов О.О., Остроух А.В. Об одном подходе к решению задачи регулирования дорожного движения: автоматизация системы управления светофорами // Автотранспортное предприятие. 2011. № 6. С. 43–45.
96. Хадиев А.М. Разработка и практическая реализация миварной машины логического вывода // Радиопромышленность. 2015. № 3. С. 79–89.
97. Чибилова М.О. Анализ подходов к построению систем поддержки принятия решений: онтологии и мивары // Автоматизация и управление в технических системах. 2014. № 1.2. С. 44–60.
98. Чибилова М.О. Необходимость добавления ограничений и прецедентов для развития миварного информационного пространства // Радиопромышленность. 2015. № 3. С. 67–78.
99. Чибилова М.О. Сравнительный анализ миварного подхода с подходами основывающимися на онтологиях и когнитивных картах // Радиопромышленность. 2015. № 3. С. 55–66.
100. Чибилова М.О. Сравнительный анализ возможностей когнитивных карт и миварных сетей для построения систем поддержки принятия решений // Автоматизация и управление в технических системах. 2014. № 1.1. С. 40–54. URL: auts.esrae.ru/8-158. DOI: 10.12731/2306-1561-2014-1-5.
101. Чибилова М.О. Структурное развитие миварного подхода: классы и отношения // Радиопромышленность. 2015. № 3. С. 44–54.
102. Чибилова М.О., Сергушин Г.С., Варламов О.О., Елисеев Д.В., Хадиев А.М. Реализация общедоступного миварного универсального решателя задач на основе адаптивного активного логического вывода с линейной сложностью и облачных технологий // Искусственный интеллект. 2013. № 3. С. 512–523.

Список литературы

103. Чиби́рова М.О., Серге́шин Г.С., Елисе́ев Д.В., Варла́мов О.О. «Облачная» реализация миварного универсального решателя задач на основе адаптивного активного логического вывода с линейной сложностью относительно правил «если-то-иначе» // Автоматизация и управление в технических системах. 2013. № 2. С. 22–38.
104. Чуви́ков Д.А. Применение графического движка в решении интеллектуальных задач, связанных с ситуационным трехмерным моделированием // Радиопромышленность. 2015. № 3. С. 200–209.
105. Чуви́ков Д.А. Применение процедурной анимации в решении интеллектуальных задач и проблем, связанных с ситуационным трехмерным моделированием // Радиопромышленность. 2015. № 3. С. 184–190.
106. Чуви́ков Д.А. Применение физического движка в решении задач, связанных с ситуационным трехмерным моделированием в реальном времени // Радиопромышленность. 2015. № 3. С. 191–199.
107. Чуви́ков Д.А., Каза́кова Н.А., Варла́мов О.О., Хади́ев А.М. Анализ технологий трехмерного моделирования и создания 3d объектов для различных интеллектуальных систем // Автоматизация и управление в технических системах. 2014. № 2.1. С. 84–97.
108. Чуви́ков Д.А., Пете́рсон А.О. Применение миварных технологий в интеллектуальном имитационном моделировании // Автоматизация и управление в технических системах. 2015. № 4.1. URL: auts.esrae.ru/16-346.
109. Чуви́ков Д.А., Феокти́стов В.П. Сравнительный анализ 3D форматов хранения данных в интеллектуальных системах и системах виртуальной реальности // Автоматизация и управление в технических системах. 2014. № 4. С. 3–14. DOI: 10.12731/2306-1561-2014-4-1.

WiMi СРЕДА РАЗРАБОТКИ МОДЕЛЕЙ ЗНАНИЙ

WiMi – инструмент для создания моделей знаний с неограниченным количеством связей, параметров и отношений.

WiMi позволяет строить сложные логические цепочки из простых правил «если ..., то ...». Для построения алгоритма используется логический вывод с линейной вычислительной сложностью.

WiMi может использоваться как для создания программных роботов, так и для построения сложных экспертных систем или логически решающих систем.

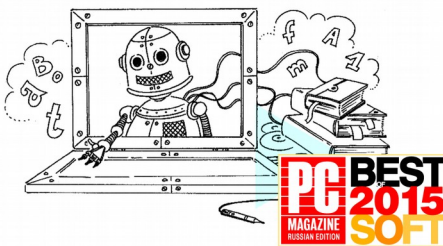
РЕШАЕМЫЕ ЗАДАЧИ:

- ◆ Построение экспертных систем
- ◆ Создание моделей знаний и алгоритмов для логически решающих систем
- ◆ Генерация алгоритмов исполнения – логический вывод

ПРЕИМУЩЕСТВА:

- ◆ Использование существующих знаний для обработки (формализация)
- ◆ Автоматическая генерация алгоритмов решения из имеющихся знаний
- ◆ Обоснование решения
- ◆ Эволюционность моделей знаний
- ◆ Скорость обработки логического вывода > 5 000 000 правил/сек.

ПОДДЕРЖИВАЕМЫЕ ПЛАТФОРМЫ:



WiMi зарегистрирован в Едином реестре российского ПО как «ЭСМИ – Конструктор Экспертных Систем Миварный». Приказ Минкомсвязи России от 28.04.2016

TelMi СЕМАНТИЧЕСКАЯ ПЛАТФОРМА

ТЕХНОЛОГИЯ ПОНИМАНИЯ ЕСТЕСТВЕННОГО ЯЗЫКА И СМЫСЛОВОГО АНАЛИЗА ТЕКСТА

Применение семантических технологий, построенных на миварном подходе, позволило преодолеть принципиальные ограничения в понимании естественного русского языка.

ПРИНЦИП РАБОТЫ:

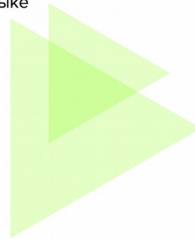
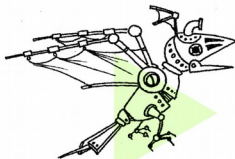
Используемые в **TELiMi** алгоритмы преобразуют текст в миварный семантический граф. Анализ текста включает синтаксический разбор, семантическое преобразование, построение графа, а также работу с контекстом и снятие неопределенностей.

База данных системы состоит из концептов (понятий), их определений, системных отношений и контекстов.

Семантический граф, а также модули платформы позволяют не просто искать слова в словаре, но и распознают контекстное окружение – тем самым достигается максимально точное понимание смысла текста.

ПРАКТИЧЕСКОЕ ПРИМЕНЕНИЕ:

- ◆ Умный поиск
- ◆ Виртуальный консультант
- ◆ Антиплагиат – поиск заимствований
- ◆ Интеллектуальное аннотирование и реферирование
- ◆ Мониторинг и фильтрация контента
- ◆ Автоматическое построение онтологий
- ◆ Восприятие роботами команд на естественном языке



РОБОРАЗУМ – ПРОГРАММНАЯ ПЛАТФОРМА ИНТЕЛЛЕКТУАЛИЗАЦИИ РОБОТОТЕХНИЧЕСКИХ УСТРОЙСТВ

**ИНТЕЛЛЕКТУАЛИЗАЦИЯ – ПОСЛЕДНИЙ ФАКТОР,
СДЕРЖИВАЮЩИЙ РАЗВИТИЕ АВТОНОМНЫХ РОБОТОВ.**

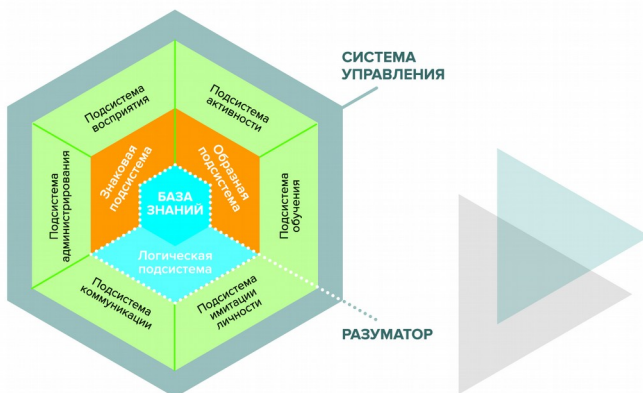
РОБОРАЗУМ – встраиваемая программная платформа для построения систем управления автономными роботами. Наделяет робототехнические устройства и комплексы искусственным интеллектом, дает им возможность действовать по-настоящему автономно и независимо от человека-оператора. Не требует мощных вычислительных ресурсов.

СОСТАВ РЕШЕНИЯ:

- ◆ Логическое ядро (РАЗУМАТОР)*
- ◆ Подсистемы автономного управления
- ◆ Драйверы

РАЗУМАТОР – это логический планировщик и ядро платформы. РАЗУМАТОР позволяет автономным роботам самостоятельно строить алгоритмы, принимать решения и решать задачи без участия людей.

**ЕСЛИ РОБОТОМ МОЖНО УПРАВЛЯТЬ ДИСТАНЦИОННО,
МЫ СДЕЛАЕМ ЕГО АВТОНОМНЫМ!**



Учебное издание

Варламов О.О., Чибирова М.О., Хадиев А.М., Антонов П.Д.,
Сергушин Г.С., Шошев И.А., Назаров К.В.

ПРАКТИКУМ ПО СОЗДАНИЮ МИВАРНЫХ
ЭКСПЕРТНЫХ СИСТЕМ

Учебное пособие

Научный редактор О.О. Варламов
Компьютерная верстка Ю.В. Горбачева
Редактор-корректор Ю.В. Горбачева
Рисунки Л.О.Варламова

Подписано к печати 17.08.2016
Формат 100x70 1/16 Бумага офсетная
Печать офсетная Тираж 300 экз.
Усл.п.л. 14,84
Уч.-изд.л. 5,16

Отпечатано с оригинал-макета заказчика
В типографии ООО «Белый ветер»
115054, Москва, ул. Щипок, 28