

специалиста по проектированию подсистемы данных КИМПО, играет разработчик приложения. Роль источника знаний о ПрО, конечного пользователя, играет заказчик приложения из структурного подразделения (как правило, это руководители различного уровня). Разработчик приложения, работая в тесном контакте с заказчиком приложения, формирует в результате описание сущностей, соответствующих требуемым исходным данным. На втором этапе роль специалиста по проектированию всей КИМПО выполняют главный специалист по проектированию, который взаимодействует с разработчиком приложения, уточняя сформированные на первом этапе описания сущностей, и обеспечивает интеграцию представленных сущностей с КИМПО.

Кроме собственного проектирования КИМПО, предлагаемая методика может быть успешно использована в учебном процессе, например, для выполнения курсовой работы студентами вуза по дисциплине «Базы данных». Ввиду того, что для практического обучения используются, как правило, упрощенные, учебные модели реальных проблемных ситуаций, в процессе обучения основные этапы проектирования должны быть пройдены за гораздо более короткий период времени, поэтому студентам предлагаются достаточно простые описания ПрО. При этом обучаемый должен освоить все роли участников процесса проектирования КИМПО. Таким образом, при выполнении курсовой работы в вузе с использованием предложенной методики все действия по проектированию выполняются одним человеком – обучаемым. Во-первых, он выполняет роль пользователя, представляющего и интерпретирующего исходные информационные потребности по рассматриваемой ПрО, определяя при этом их семантику, зависимости и возможные значения. Как проектировщик студент анализирует представленные потребности и формирует на их основе КИМПО. Следует отметить, что частично роль пользователя играет преподаватель, консультируя обучаемого в части интерпретации задания на курсовую работу.

Тот факт, что проектирование КИМПО является во многом искусством, обусловлен не тривиальностью процедур нормализации. Для эффективного повышения качества тестирования необходимо иметь возможность генерации большого количества неповторяющихся вариантов тестов, содержащих схемы ненормализованных реляционных отношений (сущностей). Для автоматической генерации тестовых заданий предлагается использовать обратную процедуру соединения нормализованных сущностей, представленных в КИМПО для получения ненормализованных сущностей, содержащих необходимые нарушения нормальных форм. Процедура формирования заданий и ответов в зависимости

от типа связей и доменов атрибутов, а также подробное описание методики проектирования КИМПО представлено в работах [1, 2].

#### Выводы

Предлагаемая интеграционная методика проектирования КИМПО позволяет автоматически выявлять связи между сущностями различного ПрО, к числу которых относятся и вузу. Применение описанной методики позволяет повысить (направлено на повышение) уровень формализации процесса моделирования конечной структуры БД, сократив при этом степень необходимого участия конечных пользователей и возможность существования неучтенных взаимосвязей. Наконец, упрощенная для использования в учебном процессе методика может быть применена при разработке КИМПО для относительно простых ПрО, а также при генерации тестовых заданий для организации контроля знаний у студентов вуза.

#### Список литературы

1. Исаев И.В. Модели и алгоритмы формирования концептуальной схемы данных единой информационной среды вуза: дисс. ... канд. тех. наук. – Томск, 2005. – 210 с.
2. Осипова В.В. Разработка и исследование метаописаний информационной базы в автоматизированных системах организационного управления: дис. ... канд. тех. наук. – Томск, 2011. – 140 с.
3. Чудинов И.Л. Базы данных: учебное пособие / И.Л. Чудинов, В.В. Осипова. – Томск: Изд-во Томского политехнического университета, 2011. – 140 с.
4. Barker R. Case\*Method–Entity Relationship Modelling. – England, Wokingham: Addison Wesley Professional, 1990.
5. Kim Y.-G. Comparing Data Modeling Formalisms / Y.-G. Kim, S.T. March // Communications of the ACM. – Vol. 38, № 6. – 1995. – P. 103-115.

### ПРИМЕНЕНИЕ МУЛЬТИАГЕНТНЫХ СИСТЕМ В ИНТЕЛЛЕКТУАЛЬНЫХ ЛОГИСТИЧЕСКИХ СИСТЕМАХ

Розенберг И.Н., Цветков В.Я.

НИИАС, Москва, e-mail: cvj7@list.ru

Основная цель интеллектуальной логистики это... процесс организации цепочки доставки и управления этой цепью на основе интеллектуальных систем и интеллектуальных технологий. Эта цель означает, что логистические процессы направлены на доставку грузов с основными ключевыми показателями: *минимальные затраты, точное время, заданное место*

Разделяют тактические и стратегические задачи интеллектуализации транспортной логистики. Стратегическая задача интеллектуализации транспортной логистики состоит в разработке интеллектуальной логистической системы (ИЛС) и решение логистических задач на этой основе. Тактическая задача интеллектуализации транспортной логистики состоит в интеграции существующих методов логистики на базе интеллектуальных технологий и комплексных методов транспортной логистики.

ИЛС – распределенная интеллектуальная система учета, регистрации, координации, кон-

троля, управления транспортными потоками и состоянием транспортной инфраструктуры, а также отношений между транспортной сферой и сферой управления [1]. Основной функцией ИЛС является решение логистических задач при условии невозможности эффективного решения их с помощью обычного человеческого интеллекта.

Основными факторами невозможности применения человеческого интеллекта это: информационная неопределенность, нестационарность внешней среды (динамическая неопределенность), информационная сложность моделей и ситуаций. В сложной ситуации единственное оптимальное решение логистической задачи найти невозможно по тем же причинам. На практике в таких случаях ограничиваются поиском не оптимальных, а достаточно «хороших», с точки зрения практики, решений

Выходом из таких ситуаций является применение методов интеллектуальной логистики, в частности, мультиагентных интеллектуальных систем. Сущность мультиагентного подхода состоит в редуцировании исходной сложной задачи в совокупность простых задач, совместное решение которых приведет к решению исходной задачи. При этом решение каждой «простой» задачи осуществляется специальной решающей системой, называемой агентом.

Агентом называют решатель задач, который представляет собой программную сущность, способную действовать в интересах достижения поставленных целей [2]. Агенты обладают рядом свойств, среди которых следует выделить:

*Автономность* – агенты функционируют без прямого вмешательства ЛПП и обладают способностью контролировать свои действия и внутреннее состояние.

*Информационное взаимодействие* – агенты взаимодействуют с другими агентами средствами некоторого коммуникационного языка.

*Реактивность* – агенты реагируют на изменения окружающей среды в определенных временных рамках.

*Целенаправленность* – агенты обладают целенаправленным поведением и способностью проявлять инициативу.

*Индивидуальность* – каждый агент имеет собственную модель окружающего его мира, на основе которой осуществляет действия.

*Коммуникабельность и кооперативность* – агенты могут обмениваться информацией с окружающей их средой и другими агентами.

*Интеллектуальность* – поведение агента включает способность к самоорганизации или конструированию модели окружающей среды для того, чтобы находить новые способы поведения.

Решение всякой простой задачи может быть представлено в виде продукции:

$$K_p \rightarrow K_T, \quad (1)$$

где  $K_p$  – модель реального состояния объекта;  $K_T$  – модель требуемого состояния объекта. Решение задачи может быть расчлнено на отдельные действия решающей системы и в целом представлено как последовательность этих действий:

$$K_p \rightarrow d1(K_p) \rightarrow K_1 \rightarrow \\ \rightarrow d2(K_1) \rightarrow K_2 \rightarrow d3(K_2) \rightarrow K_3 \rightarrow \dots \rightarrow K_T, \quad (2)$$

Последовательность действий решающей системы  $\langle d_1, d_2, \dots, d_n \rangle$  есть путь решения задачи. Под путем решения часто понимают алгоритм решения задачи. По этому критерию все задачи можно разделить на два типа. Если путь решения исходной задачи известен априори, то имеет место решающая система первого рода (1). Если же путь решения исходной задачи неизвестен, то решающая система называется решающей системой второго рода.

Для решающих систем второго рода путь решения должен быть задан неявно. Для таких систем характерно, что для них способом решения задач всегда является некоторый метод поиска последовательности  $\langle d_1, d_2, \dots, d_n \rangle$  выражение (2).

В общем, задача поиска решения формулируется следующим образом. Пусть заданы множества начальных и целевых состояний ( $K_p, K_T$ ) и множество операторов  $d$ . Необходимо найти упорядоченный набор действий решающей системы  $\langle d_1, d_2, \dots, d_n \rangle$ , такой, который переводит множества начальных состояний  $K_p$  приводит в требуемые целевые состояния  $K_T$ .

Именно такую задачу решают агенты. Мультиагентная система (МАС) – это решающая система, включающая совокупность агентов для получения решения задачи.

Использование мультиагентного подхода позволяет решить задачи более сложные, чем те, которые могут быть решены при помощи классических экспертных систем, например, задачи, обладающие неопределенностью по статической структуре оригинала. Как и многие сетевые структуры, коллективы агентов могут быть гомогенными, когда все агенты коллектива имеют однотипную структуру, и гетерогенными, в противном случае. Как сложная система коллектив агентов имеет элементы, которыми являются агенты.

Эти элементы могут быть связаны пространственными отношениями, например, функционировать на разных пространственных узлах, или временными отношениями порядка, например, одни агенты начинают работать только после окончания работы других агентов.

Как сложная система коллектив агентов обладает свойством полноты и иерархичности. Полнота означает возможность решения любой исходной задачи из проблемной области. Иерархичность означает структурированность по уровням обрабатываемой агентами информации.

Применение мультиагентных интеллектуальных систем позволяет решать задачи с динамической неопределенностью, информационной

неопределенностью и информационной сложностью (применительно к человеческому интеллекту). Этим расширяются границы применимости интеллектуальных логистических систем и интеллектуальных транспортных систем.

**Список литературы**

1. Розенберг И.Н., Цветков В.Я. Интеллектуализация транспортной логистики // Железнодорожный транспорт. – 2011. – № 4. – С. 38–40  
 2. Иванников А.Д., Кулагин В.П., Тихонов А.Н., Цветков В.Я. Геоинформатика. – М.: Макс Пресс, 2001. – 349 с.

**О ТЕРМОДИНАМИКЕ РАСШИРЕНИЯ ГАЗОВОГО ПУЗЫРЯ**

Снопов А.И., Гетманский М.С.

*Южный федеральный университет,  
 Ростов-на-Дону, e-mail: asnop@math.rsu.ru*

Исследованию динамики и термодинамики пузырей в жидкостях, в последние годы уделяется значительное внимание в связи с возросшей ролью этого явления в технике и достижениями современной технологии вычислений [1-3]. В частности показано, что при сжатии пузырьков микронных размеров температуры в них могут достигать десятков тысяч градусов.

Ниже представлены результаты расчета термодинамики расширения пузыря, образованного с помощью электролиза в воде, фотографии которого представлены в книге [4]. Сведений о термодинамических параметрах пузыря в ней нет. Ее автор ограничился только приближенным расчетом в рамках теории кавитации зависимости от времени радиуса сжимающейся каверны. Однако после схлопывания опять образовался пузырь, который изображен на фото в процессе расширения, что позволяет провести теоретическое исследование его внутренней термодинамики. Согласно фото рост среднего радиуса пузыря  $r$  достаточно хорошо описывается формулой

$$r = r_0 \sqrt{1 + 2bt} \quad (1)$$

при  $r_0 = 0,0047$  м и  $b = 35000$  с<sup>-1</sup>.

Для случая расширения газового объема в соответствии с формулой (1) в работе [3] представлено точное решение уравнений динамики и термодинамики вязкого газа, которое в переменных Лагранжа имеет вид

$$\begin{cases} \rho = \rho_0 (1 + 2bt)^{-\frac{3}{2}} \\ p = 0,5\rho_0 \xi^2 b^2 (1 + 2bt)^{-\frac{5}{2}} + C(1 + 2bt)^{-\frac{3k}{2}} \\ T = p / (R \rho) \end{cases} \quad (2)$$

Здесь  $\rho$  – плотность;  $p$  – давление;  $T$  – температура;  $R$  – газовая постоянная;  $\xi$  – расстояние от центра пузыря, притоки тепла к пузырю не учтены.

Из условия равенства давления в пузыре в конце его расширения атмосферному давлению определяется значение величины  $C$ , что дает возможность рассчитать поля давлений, плотностей и температур в газовом пузыре с момента его возникновения. На результаты расчетов существенно влияют величины параметров  $R$  и  $k$ . Например, при  $R = 519$  и  $k = 1,33$  имеем  $p_0 = 1,72 \cdot 10^9$  Н/м<sup>2</sup>,  $T_0 = 3300$  К, а для  $k = 1,67$  находим  $p_0 = 2,09 \cdot 10^9$  Н/м<sup>2</sup>,  $T_0 = 40200$  К. Начальная скорость границы облака составляет 16,45 м/с, конечная – 1,45 м/с. Температура в облаке согласно расчетам в этот момент составляет 293 К.

Представленные расчеты показывают, что модель расширения газового облака [3] может быть использована для анализа термодинамики расширения пузырей в жидкости.

**Список литературы**

1. Динамика пузырька газа в центре сферического объема жидкости / А.А. Аганин, Р.И. Нигматуллин, М.А. Ильгамов и др. // Докл. РАН. – 1999. – Т. 369, №2. – С. 182–185.  
 2. Аганин А.А., Халитова Т.Ф., Хизматуллина Н.А. Расчет сильного сжатия сферического парового пузырька в жидкости // Вычислительные технологии. Институт вычислительных технологий СО РАН. – Новосибирск. – 2008. – С. 17–27.  
 3. Снопов А.И. Расширение (сжатие) конечного объема вязкого газа // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2011. – № 4. ч. 3. – С. 1120–1121.  
 4. Бэтчелор Дж. Введение в динамику жидкости // М.: Мир, 1973. – 792 с.

**РАЗРАБОТКА ФРЕЙМВОРКА ДЛЯ СОЗДАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ЛЕЧЕБНО-ДИАГНОСТИЧЕСКИМ ПРОЦЕССОМ**

Таранов Ю.А., Борзых В.Э.

*Тюменский государственный нефтегазовый университет, Тюмень, e-mail: taranovalv@list.ru*

Информационные системы для сбора, обработки, хранения и анализа данных используются во всех отраслях, в т.ч. и в медицине. В связи с тем, что в медицине требуется оперативная обработка информации и незамедлительное реагирование на происходящие события, постоянно актуальна проблема сбора и анализа данных. Медицинские информационные системы создаются, как правило, для нужд конкретного лечебно-профилактического учреждения с учетом его специфики. Для создания единого информационного пространства медицинских учреждений необходима система, которая может быть использована для обмена информацией в требуемом и единообразном виде.

В этой связи в настоящей работе предприняты шаги к созданию полноценной базы для разработки медицинских АСУ. Основная идея – разработать фреймворк, который позволил бы легко и быстро создавать автоматизированные информационно-аналитические системы для различных лечебных учреждений. За основу был