

ИЗУЧЕНИЕ ФИЗИКО-МЕХАНИЧЕСКИХ СВОЙСТВ ЦЕМЕНТНЫХ СИСТЕМ НА ОСНОВЕ ВТОРИЧНЫХ РЕСУРСОВ ПТК «ХИМВОЛОКНО»

Н.А. ШЕЙБАК, О.В. БИЛИДА, Д.И. САФОНЧИК

The paper studies the feasibility of using recycled sodium nitrite generated during cleaning of process equipment for the production of polyamide fibers in a PTC "Grodno Chemical Fibre" as a modifying additive to cement concretes and mortars

Ключевые слова: добавки, нитрит натрия, прочность

Применению химических добавок в технологии бетонных работ в мировой практике уделяется огромное внимание. Учитывая этот факт, в проделанной работе ставилась задача определить возможность применения в качестве модифицирующей добавки в бетоны и растворы вторичного сырья, образующегося при производстве полиамидного волокна на предприятии ПТК «Гродно Химволокно». Данное вторсырьё ввиду не изученности его свойств, в т.ч. и влияния на цементные системы, нигде не используется и занимает огромную площадь складских помещений [1].

Схема получения вторичного нитрита натрия в условиях ПТК «Гродно Химволокно» представлена на рисунке 1.

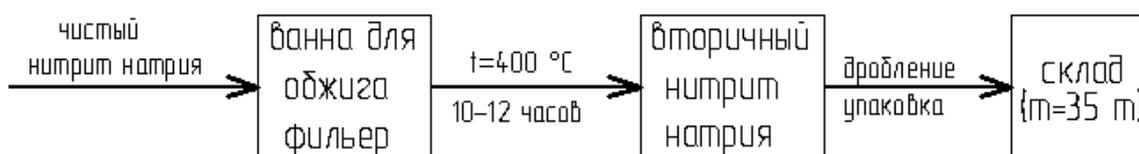


Рис. 1 – Схема получения вторичного нитрита натрия

Используя полученное экспериментальным путём водоцементное отношение, были приготовлены образцы-кубики с ребром 7,07 см для испытания на центральное осевое сжатие в различном возрасте твердения. Для эксперимента были использован портландцемент ПЦ 500-Д20, соль карбоната натрия.

Введение добавки NaNO_2 в количестве 3–5 % от массы цемента, не увеличивает относительной прочности образцов при твердении в нормальных условиях и даже несколько её снижает. При этом абсолютная прочность увеличивается, но менее интенсивно, чем в варианте без добавки.

При модификации растворов добавкой Na_2CO_3 в количестве 3–5 %, скорость набора прочности по отношению к бездобавочному варианту возрастает на протяжении 1–3 суток. Однако на седьмые сутки наблюдается значительное снижение темпов твердения.

При введении в растворную смесь добавки вторичного нитрита натрия в количестве 3–5 %, скорость набора прочности по отношению к бездобавочному варианту также возрастает на протяжении первых 3 суток. Более высокая прочность наблюдалась и при твердении в возрасте 7 суток.

Проводились испытания на прочность образцов, твердевших при минус 5 °С. Полученные результаты свидетельствуют о том, что при твердении образцов с 3 % кальцинированной соды в условиях отрицательной температуры (минус 5 °С) скорость набора прочности происходит очень медленно. Введение соды в качестве ускорителя твердения при отрицательной температуре является нецелесообразным.

Вторичный нитрит натрия позволяет увеличивать скорость набора прочности растворов в начальный период твердения, особенно это проявляется при твердении в условиях низких положительных температур.

Литература

1. Экологические и ресурсосберегающие технологии промышленного производства: сборник статей международной науч.-техн. конф., Витебск, 24–25 октября 2006 г. / УО «ВГТУ»; редкол.: П.А. Витязь [и др.]. – Витебск, 2006. – С. 183–185.

СЕМАНТИЧЕСКАЯ ПАРАЛЛЕЛЬНАЯ АСИНХРОННАЯ МОДЕЛЬ РЕШЕНИЯ ЗАДАЧ

Д.В. ШУНКЕВИЧ

This work is devoted to considering of problems of existing methods, means and technologies of intelligent problem solvers design. The article describes the problem of lack of means, which allow relatively inexperienced developer to design of intelligent problem solver for applied intelligent systems for different purposes. Next part of the paper considers the technology which was designed to solve given problem in the way of integration different problem situations solution methods and ways on the common formal basis

Ключевые слова: интеллектуальная система, интеллектуальный решатель задач, многоагентная система, агент

1. ВВЕДЕНИЕ

Целью современных программных систем в рамках решения поставленной задачи, как правило, является получение какого-либо результата. При этом сам процесс достижения этого результата обычно понятен только компьютеру и по этой и другим причинам просто скрывается от пользователя. Но во многих случаях существует необходимость анализа непосредственно решения, исследования пути получения ответа в каждой задаче. При этом от системы требуется объяснение понятное не только ей, но и пользователю, причем, как правило, пользователем здесь выступает лицо без специального образования, позволяющего с легкостью ориентироваться в процессах, происходящих внутри информационной системы. Наиболее актуальна такая возможность для класса обучающих систем, основной целью которых является именно объяснение каких-либо действий, предпринимаемых в некоторой проблемной ситуации. Непосредственно ответ на поставленную задачу здесь имеет значительно меньшую ценность, чем способ его получения.

Если отойти от рассмотрения только обучающих систем, то выясняется, что не меньшую актуальность имеет проблема универсализации решения различных задач в различных предметных областях. Существующие программные средства, как правило, решают задачи по жестко заданным алгоритмам. Задание способов решения для каждого вида задач осуществляется разработчиком на этапе проектирования. Данный факт накладывает ряд ограничений на последующее использование системы:

- добавление в систему нового функционала, позволяющего решать большее количество задач, либо оптимизирующего решение уже существующих, не может быть осуществлено самим пользователем, либо требует от него достаточно трудоемкого и длительного обучения специальным средствам (как например, изучение языка Visual Basic для написания макросов в пакете MS Office). Чаще всего в таких случаях пользователю прикладной системы проще всего обратиться к ее разработчикам и за дополнительную плату попросить внести требуемые изменения;

- Процесс проектирования системы требует от разработчика фактически перечисления всевозможных видов задач, которые должна решать система с указанием действий в каждом конкретном случае. Это, во-первых, увеличивает трудоемкость проектирования, и, во-вторых, требует от разработчика совмещения навыков программирования и теоретических знаний о различных алгоритмах решения задач.

Еще одной немаловажной проблемой существующих подходов к созданию информационных систем является проблема кроссплатформенности и переносимости. Существует много различных языков программирования, операционных систем и аппаратных средств, а проектирование конкретной системы зависит от каждой из трех указанных компонент. Проблема зависимости программной системы от аппаратной реализации вычислительного устройства на данный момент в значительной мере решена на уровне операционных систем и языков программирования высокого уровня. Существуют также языки высокого уровня (такие как Java, Python и многие другие), позволяющие обеспечить переносимость программной системы между различными операционными системами. Однако на настоящее время не существует средств, позволяющих в полной мере исключить проблемы, возникающие при переносе программной системы, реализованной на произвольном языке программирования на различные операционные системы. Попытки решения проблемы кроссплатформенности предпринимаются с 1989 года консорциумом Object Management Group (OMG). Предлагаемый подход базируется на концепции Model-Driven Architecture, основным принципом которой является замена привычного программирования описанием модели проектируемой системы на некотором метаязыке (например, UML) и задание правил трансформации модели на различные платформы. [1]

Тесно связана с проблемой переносимости проблема интеграции различных видов знаний, в частности различных алгоритмов и методов решения задач. Процесс совмещения работы систем, реализованных на различных языках программирования очень трудоемок даже для таких языков высокого уровня, построенных на схожих принципах, как Java и C#.

Целью данной работы является решение части из описанных выше проблем путем создания универсальной семантической модели решения задач, которая должна удовлетворять следующим свойствам:

- Универсальность. Проектируемая модель должна обеспечивать возможности для решения произвольных классов задач в различных предметных областях, не требуя при этом вмешательства пользователя данной разработки в ее внутренне устройство.

- Модульность и расширяемость. Проектируемая модель решения задач должна обеспечивать возможность расширения функционала системы, в которой она используется, без изменения самой модели.

- Кроссплатформенность. Проектируемая модель решения задач не должна зависеть от операционной системы и аппаратной архитектуры устройства, на котором предполагается работа информационной системы.

- **Параллельность.** Проектируемая модель должна обеспечивать возможность параллельного использования различных способов решения задач в рамках решения одной задачи, а также возможность параллельного решения сразу нескольких задач. При этом необходимо обеспечить согласованность и интегрируемость результатов применения различных методик решения задач.

Наиболее актуальным будет использование проектируемой параллельной асинхронной модели решения задач при создании интеллектуальных обучающих систем по различным предметным областям.

Для достижения поставленной цели в рамках данной работы необходимо решить следующие задачи:

- проанализировать предметную область, связанную с параллельной асинхронной обработкой информации, а также моделями решения задач в интеллектуальных системах;
- спроектировать семантическую модель параллельного асинхронного решения задач в рамках семантической технологии компонентного проектирования интеллектуальных решателей задач;
- реализовать спроектированную модель решения задач с использованием специализированного языка обработки семантических сетей SCP [2];

Результатом работы в данном направлении будет модель параллельного асинхронного решения задач в рамках семантической технологии компонентного проектирования интеллектуальных решателей задач в проекте OSTIS. Данная модель позволит интегрировать различные стратегии решения задач и методы логического вывода в рамках прикладных интеллектуальных справочных систем по различным предметным областям. Наличие такой модели позволит значительно упростить и ускорить проектирование интеллектуальных решателей задач для подобных систем, а использование средств параллельной обработки при решении задач позволит в будущем перенести прикладную интеллектуальную систему на распределенную архитектуру. Это позволит существенно улучшить производительность такой системы при условии наличия распределенной системы хранения знаний в виде ассоциативной графодинамической памяти.

Основой для интеграции различных видов знаний, в том числе и моделей решения задач, является представление знаний на основе унифицированного языка описания семантических сетей. [3]

2. ПРОЕКТИРОВАНИЕ СЕМАНТИЧЕСКИХ ПАРАЛЛЕЛЬНЫХ АСИНХРОННЫХ МОДЕЛЕЙ РЕШЕНИЯ ЗАДАЧ

2.1 Унифицированные модели решателей задач, разработанных на основе технологии OSTIS

В предлагаемом подходе к преодолению приведенных проблем решатель задач рассматривается в неклассическом варианте. В данном случае решатель задач представляет собой графодинамическую sc-машину (память в качестве модели представления знаний использует семантическую сеть), состоящую из двух частей:

- графодинамическая sc-память;
- система sc-операций (sc-агентов).

Система операций является агентно-ориентированной и представляет собой набор sc-операций, условием инициирования которых является появление в памяти системы некоторой определенной конструкции. При этом операции взаимодействуют между собой через память системы посредством генерации конструкций, являющихся условиями инициирования для другой операции. При таком подходе становится возможным обеспечить гибкость и расширяемость решателя путем добавления или удаления из его состава некоторого набора операций.

Отличительной особенностью решателя задач как многоагентной системы в рамках данного подхода является принцип взаимодействия операций-агентов. По сути, предлагаемый подход реализует принцип «доски объявления», рассматриваемый в теории MAC. Агенты обмениваются сообщениями исключительно через общую память путем использования соответствующего языка взаимодействия (языка вопросов-ответов), в отличие от большинства классических MAC, в которых агенты обмениваются сообщениями непосредственно друг с другом. В рассматриваемом подходе каждый агент, формулируя вопросную конструкцию в памяти, априори не знает, какой из агентов будет обрабатывать указанную конструкцию, а лишь дожидается появления в памяти факта окончания обработки вопроса. При этом в решении поставленной таким образом задачи может принимать участие целый коллектив агентов. Аналогичным образом, реагируя на появление некоторой конструкции в памяти, агент в общем случае не знает, кто из его коллег поставил данный вопрос, а лишь может проверить соответствие сгенерированной конструкции своему условию инициирования. В случае наличия такого соответствия, агент начнет обработку указанного вопроса (решение поставленной задачи), и в результате работы сгенерирует некоторый ответ на поставленный вопрос.

Проверка соответствия сгенерированного вопроса условиям инициирования агентов происходит следующим образом: автору вопроса после его формулирования необходимо инициировать данный вопрос (включить его во множество инициированных вопросов). После инициирования вопроса каж-

дый из агентов, работающих в памяти, переходит в активное состояние и начинает проверку условия инициирования. При этом проверка начинается с наиболее уникальных фрагментов условия (например, типа вопроса) с целью оптимизации данного процесса. В случае установления факта изоморфности вопросной конструкции и условия инициирования агент начинает решение поставленной задачи, в противном случае агент переходит в состояние пассивного ожидания.

Описанная модель взаимодействия агентов в общей памяти позволяет обеспечить максимальную расширяемость системы агентов и предельно упростить процесс добавления новых агентов в уже имеющийся коллектив.

2.2 Иерархическая структуризация параллельной асинхронной модели решения задач

Для определения структуры рассматриваемой модели решения задач рассмотрим более подробно процесс решения задачи интеллектуальной системой.

Процесс решения задачи можно разделить на следующие этапы:

- Этап работы поисковых операций.

Вне зависимости от типа задачи всегда имеется вероятность того, что данная задача уже была решена системой ранее или системе уже откуда-либо известен ответ на поставленный вопрос. На данном этапе работу осуществляет коллектив поисковых операций, каждая из которых, как правило, соответствует некоторому классу решаемых задач. Если ответ найден, решатель прекращает свою работу. В противном случае происходит переход на следующий этап решения.

- Этап применения стратегий решения задач.

На данном этапе осуществляется выбор между различными стратегиями решения задач, и, при необходимости, параллельный запуск различных стратегий. Целью работы каждой из стратегий является получение набора пар, связывающих некоторое множество объектов и логическое утверждение из БЗ, которое справедливо для классов, которым принадлежат эти объекты в рамках некоторой теории. Впоследствии при рассмотрении каждого утверждения осуществляется попытка применить его в рамках некоторой семантической окрестности рассматриваемых объектов, для чего осуществляется переход на следующий этап решения.

- Этап применения правил логического вывода.

На данном этапе происходит попытка применения утверждения, полученного на предыдущем шаге, с целью генерации в системе новых знаний. Если такое применение справедливо (например, посылка истинна) и имеет смысл (в результате применения будут сгенерированы новые знания), то осуществляется генерация новых знаний на основе одного из правил логического вывода. При этом применение происходит в контексте объекта, рассматриваемого на предыдущем этапе (в общем случае – ряда объектов). Если в данном контексте вывод на основе данного утверждения невозможен или нецелесообразен, решение возвращается на предыдущий этап. В случае успешного применения утверждения происходит переход к следующему этапу решения.

- Этап верификации и оптимизации сгенерированных знаний и сборки мусора.

На данном этапе происходит интерпретация арифметических отношений, сгенерированных в процессе решения на предыдущем этапе, то есть попытка вычисления недостающих значений компонентов связок арифметических отношений (например, сложение величин и произведение величин) на основе имеющихся значений. Если вычислить все недостающие значения не представляется возможным, то все знания, сгенерированные на предыдущем этапе, уничтожаются и решение переходит на этап применения стратегий. В таком случае применение логического вывода для рассматриваемого на предыдущем шаге утверждения считается не целесообразным. Также на данном этапе происходит устранение синонимии, если таковая появилась на предыдущем этапе решения, например, сгенерирована связка отношения совпадения между некоторыми объектами. В конечном итоге происходит удаление конструкций, ставших ненужными и по каким-либо причинам не удаленных на предыдущих этапах решения.

Если все этапы решения выполнены успешно, то решение возвращается к первому этапу, и в случае, если ответ не получен, процесс повторяется еще раз. Стоит отметить, что в процессе решения один и тот же объект или одно и то же высказывание могут быть использованы многократно, если это целесообразно. Однако, очевидно, что применение одного и того же утверждения для одного объекта несколько раз не имеет смысла, при условии, что нужные знания из памяти не удаляются в процессе решения какими-либо сторонними операциями.

Таким образом, в структуре описываемой модели можно выделить 4 логических уровня, на каждом из которых возможно использование методов параллельной обработки информации:



Рисунок 1 – Структура параллельной асинхронной модели решения задач

Литература

1. Консорциум OMG [Электронный ресурс]. – Режим доступа: <http://omg.org/>. – Дата доступа: 20.03.2012.
2. Голенков, В. В. Программирование в ассоциативных машинах / Голенков В. В. [и др.]; под ред. В. В. Голенкова – Минск, БГУИР, 2001 – 276 с.
3. Голенков, В. В. Представление и обработка знаний в графодинамических ассоциативных машинах / Голенков В. В. [и др.]; под ред. В. В. Голенкова – Минск, БГУИР, 2001. – 412с.

©ПГУ

ПРИМЕНЕНИЕ ПРИСАДОК В ДИЗЕЛЬНЫХ ДВИГАТЕЛЯХ

В.В. ЩЕПЕЛЬКОВ, Л.И. ЛАГУН

The heart of the diesel engine is its fuel injection equipment and the major enemy of this equipment is poor quality diesel fuel. Poor quality diesel fuel can cause performance issues, premature wear, gumming of components, and plugged filters

Ключевые слова: топливо, дизель, присадки

Дизельные двигатели стали очень популярны. Главная особенность эксплуатации дизелей заключается в том, что работают они на нашем топливе. А оно сильно отличается от европейского по ряду параметров: по цетановому числу, содержанию серы и др. Таким образом, использование отечественного топлива приводит к снижению мощности, ухудшению экономичности и экологичности, а также, к существенному снижению ресурса двигателей.

Условия хранения и транспортировки дизельного топлива могут быть неудовлетворительными, да и само дизельное топливо может оказаться некачественным изначально. Известно два способа реше-