Универсальная модель построения решателей задач

Шункевич Д.В.
Кафедра интеллектуальных информационных технологий
Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
e-mail: shu.dv@bsuir.by

***Аннотация*—В работе рассматривается универсальная модель построения решателей задач интеллектуальных систем на основе семантических сетей.**

Ключевые слова: интеллектуальная система; база знаний; интеллектуальный решатель; методика проектирования; семантические сети

I. Введение

Современные компьютеры способны решать множество различных классов задач разной сложности и трудоемкости, обеспечивая при этом значительную производительность и точность, что позволяет экономить материальные и людские ресурсы. Однако, целью даже 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. 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.

Данные факты приводят к необходимости 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.

II. 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) [1].

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 МАС. Агенты обмениваются сообщениями исключительно через общую память путем использования соответствующего языка взаимодействия (языка вопросов-ответов), в отличие от большинства классических МАС, в которых агенты обмениваются сообщениями непосредственно друг с другом [2]. 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 каждый 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.

III. Универсальная модель построения решателей задач

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 тоже 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 [3].

IV. Заключение

В работе приведена методика проектирования интеллектуальных решателей задач, основанная на универсальной модели построения решателей, обеспечивающая возможность параллельного асинхронного решения задач.

Результаты, приведенные в работе, апробируются в рамках открытого проекта OSTIS [4].

[1] 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.

[2] Тарасов, В. Б. От многоагентных систем к интеллектуальным организациям / В.Б. Тарасов; – М. :Изд-во УРСС, 2002 – 352 с.

[3] Заливако С. С. Семантическая технология компонентного проектирования интеллектуальных решателей задач / С. С. Заливако, Д. В. Шункевич // Материалы международной научно-технической конференции «Открытые семантические технологии проектирования интеллектуальных систем» – Минск, 2012. С. 297 – 314.

[4] Проект OSTIS [Электронный ресурс]. Минск, 2012. – Режим доступа: http://ostis.net/. – Дата доступа: 11.06.2012