

УДК 519.7

В.Ю. Мейтус

ПРОГРАММИРОВАНИЕ И ИНТЕЛЛЕКТ

Работа посвящена проблеме создания программ, решающих задачи, которые связаны со знаниями о внешней среде и использующих эти знания. Такие программы называются интеллектуальными применительно к определению интеллекта, данному в работе. Предлагается категорная модель интеллектуальной программы в виде функтора из категории, представляющей среду, в категорию, определяющую возможные способы решения задачи.

Введение

Одним из определяющих понятий современной цивилизации является понятие «технологии», как неизменного фактора, связанного с большинством социальных и производственных процессов, протекающих в окружающей нас реальности. Технологичность – одна из важнейших характеристик современного общества, неотъемлемая составляющая нашей жизни, символ прогресса индустриальной цивилизации.

Но любая технология определяется последовательностью действий, порядок осуществления которых управляется некоторой логикой. Другими словами, в основе технологического описания лежит **программирование**, рассматриваемое как *искусство представления процессов достижения результата, с помощью заданной системы операторов и логических правил вычисления порядка их действий*. Программирование, с одной стороны, неразрывно связано с пониманием и возможностью воспроизводства любого процесса, а с другой, – с эффективным, конструктивным представлением данного процесса в виде заданных системных компонент.

Операторы и логика – составляющие элементы любой программы. Под операторами понимаются определенные действия, операции, преобразования, которые могут выбираться и варьироваться в зависимости от выполнения или невыполнения некоторых логических условий, связанных с данным процессом. Эти логические условия анализируют состояние внешней среды, в которой осу-

ществляется запрограммированный процесс, и регулируют его выполнение.

В этом смысле можно говорить о компьютерном, социальном, политическом, экономическом или производственном программировании, определяющем возможность представления и моделирования процессов, относящихся к любому из этих направлений общественного бытия. В настоящее время наиболее популярным и развитым является программирование для компьютеров, с его различными направлениями, вариантами и модификациями [1–3].

Но в последние десятилетия часто говорят о психологическом программировании, ситуационном программировании, связанным с ситуационным управлением [4], программировании социальных процессов (например, нейролингвистическое программирование [5]), программировании поведения человека. Что касается производства, то само задание производственной технологии обычно осуществляется в виде некоторой последовательности действий с использованием соответствующего оборудования, т. е. в виде программы, выполнение которой реализует процесс получения определенного продукта. С помощью языков вида UML программируются и процессы моделирования систем.

Поэтому можно считать, что **программирование** как область человеческой деятельности является *универсальным и эффективным средством эффективного представления процессов, протекающих в окружающей нас внешней среде, и, одновременно, является методом познания этих процессов*. Здесь «эффективность

представления» понимается в смысле моделирования и выполнения данных процессов с помощью существующих в распоряжении программиста средств, например, с помощью компьютеров.

Программируя процессы, мы глубже понимаем их, получаем возможность воздействовать на них, вмешиваться и изменять течение процессов, определять на уровне моделирования свойства этих процессов, возможные варианты их развития и трансформации. Если эти свойства в меньшей степени существенны для вычислительных программ, хотя и там существуют свои особенности, то для изучения социальных процессов программирование создает новый аппарат описания и исследования этих процессов, дает возможность прогнозировать их развитие, предсказывать возможные результаты течения таких процессов в зависимости от условий их протекания.

Рассмотрим детальнее компьютерное программирование, а поскольку далее речь идет только об этом виде программирования, то слово «компьютерное» будем опускать.

Для задания операторов и представления данных, с которыми они работают, в программировании используются языки программирования, которые являются средством, понимаемым как человеком, использующим языки для представления алгоритмов, так и компьютером, переводящим языковые конструкции в машинные команды.

Само развитие программирования связано с постепенным усложнением внутренней структуры данных и операторов, используемых в представлении программ. Совмещение форм представления данных и команд компьютера, найденное создателями вычислительных машин, явилось **первым революционным шагом** в развитии компьютерной техники и программирования.

Второй революционный шаг – это выделение в один универсальный программный комплекс совокупности стандартных операций и интерфейсных представлений, необходимых большинству программ, которое выразилось в создании

операционных систем как неотъемлемой части программного обеспечения современного компьютера.

Третий революционный шаг – это понимание роли структур данных и включение их как равноправной части объекта в объектно-ориентированном программировании. Управление объектом осуществляется с помощью методов (процедур и функций), также включенных в структуру объекта. При этом объекты сами могут быть составляющими структур данных, таких как массивы, записи, множества, параметры процедур и функций, списки и деревья.

Объекты хорошо вписываются в структуру современных операционных систем, используя особенности их работы. При этом достоинства и возможности объектов заранее ограничены. Даже если создавать объекты, имеющие сложную структуру и содержащие большое число полей и методов, то в любом случае нельзя рассчитывать, что они смогут правильно отразить свойства различных изменяющихся сред, в которых эти объекты могут существовать. А главное, что в них не удастся задать динамически изменяющиеся связи с другими объектами той же среды, т. е. отразить переменную структуру внешней среды, необходимую для моделирования и отражения данной среды в процессах, представляемых с помощью программ.

Одним из возможных выходов из сложившейся ситуации является наделение программ тем основным свойством, которое обеспечивает человеку успешное восприятие постоянно преобразующейся окружающей среды, приспособление к ее условиям и эффективное функционирование в этой среде. Это свойство называется **интеллектом**. До сих пор представление об интеллекте чаще всего связывали с человеком или высокоорганизованными животными. Однако, соединение возможностей интеллекта с компьютерными программами, осуществляемое не на уровне программирования, а на уровне действия самих программ, позволит создавать интеллектуальные программы, которых так не хватает сегодня.

В настоящей работе рассматривается одно из направлений развития современного компьютерного программирования, связанное с возможностями создания интеллектуальных программ и формализацией этого процесса.

Интеллект

В современной обширной литературе, посвященной интеллекту и искусственному интеллекту, как способу моделирования тех или других сторон естественного интеллекта, уже существует своя классификация основных направлений исследования. В такой классификации рассмотрим следующие направления.

1. Определение на основе сравнения с качествами, присущими человеку и связываемыми с его мышлением.

Интеллект – ум, рассудок, разум, способность мыслить, устойчивая структура умственных способностей человека, набор заданных умственных функций (сравнения, абстракции, прогнозирования, создания понятий, суждений, логических выводов).

2. Определение на основе классов выполняемых процессов или осуществляемых функций.

Интеллект как совокупность универсальных процедур, позволяющих строить конкретные алгоритмы решения частных творческих задач (Г.С. Пospelov). **Интеллект** как способность запоминать и предвидеть различные аспекты внешнего мира, включая язык, математику, физические свойства объектов, социальные ситуации [6]. **Интеллект** – весьма общая умственная способность, которая включает возможность делать заключения, планировать, решать проблемы, абстрактно мыслить, понимать сложные идеи, быстро учиться и обучаться [7].

Интеллект как выполнение общих функций, ассоциируемых с деятельностью человека. Сюда относятся:

- абстрактное мышление,
- приспособление к социальной среде и к другим средам,
- решение задач и проблем,
- распознавание образов,

- понимание смысла, выраженного предложениями и текстами,
- моделирование событий и ситуаций, определяющих некоторое поведение,
- формирование понятий, суждений, аналогий.

3. Определение, опирающееся на использование знаний, как основы представления и моделирования окружающей среды.

Интеллект – это способность на основании имеющихся знаний строить модели окружающей среды и использовать их для прогнозирования будущих изменений, состояний, преобразований, поведения. Под интеллектом системы понимается её способность формировать, сохранять и использовать знания, образующие базу знаний системы, при определении взаимодействия системы с внешней средой [8].

Кроме этих определений интеллект связывают с выполнением таких человеческих функций, как распознавание образов, речи, понимание естественного языка, смысла высказываний, или с моделированием функций нервной системы с помощью нейронных сетей [9, 10]. Исследуя результаты психологических экспериментов методами факторного анализ, Р. Стернберг выделил три формы интеллекта: вербальный интеллект (запас слов, эрудиция, умение понимать прочитанное); способность решать проблемы; практический или социальный интеллект (умение адаптироваться к социальной обстановке, добиваясь поставленных целей).

Данное далее определение интеллекта основано на формулировке свойства, характерного для человека, который постоянно решает различные задачи.

Интеллект – это качество субъекта, ориентированного на взаимодействие с внешней средой, которое позволяет ему виртуально структурировать внешнюю среду, для того, чтобы, сохранив полученное структурное представление, использовать его для адекватного восприятия, моделирования и взаимодействия с этой средой, включая принятие решений при формировании своих действий [11].

Прежде всего подчеркнем, что интеллект проявляется во взаимодействии с окружающим миром. Это значит, что субъект получает сигналы и информацию из внешней среды, реагирует на них, строя для себя модель среды. Такая модель для субъекта – основа его поведения в среде.

Структурирование среды предполагает, во-первых, выделение в ней множества элементов и возможных связей между ними. Это – *анализ* среды. Во-вторых, агрегирование данных элементов между собой в более общие объекты, существующие в среде, за счет отождествления связей между ними (*синтез*). Такие процессы неоднозначны, поскольку они аккумулируют, включают в себя опыт, представления и знания тех субъектов, которые выполняют структурирование.

В приложении к взаимодействию со средой процесс структурирования ориентирован на «вырезание» из среды области, непосредственно относящейся к задаче или проблеме, которую должен решить в этой области интеллектуальный субъект. Для сохранения построенной структуры используется созданная для среды база знаний, оснащенная механизмом пополнения структуры – добавления, возможно гипотетического, новых связей и отношений. При этом даже вырезанная из среды ее часть, используемая для решения задачи, будет испытывать на себе влияние своего окружения, которое лежит вне границ данной области.

Существенно входящее в определение интеллекта понятие «адекватности восприятия, моделирования и взаимодействия с внешней средой». Именно здесь скрываются те качества, которые связываются в обычном понимании с интеллектом. Структурируя внешнюю среду, мы получаем возможность моделировать в среде действия, ассоциируемые субъектом с этой структурой. А адекватность предполагает, что эти действия не противоречат процессам, характеризующим среду, что эти действия соответствуют реальности.

Кроме того, отметим, что предлагаемое определение ориентировано на произвольные внешние среды, хотя произвольность среды ограничена возможно-

стями субъекта. Допускается, что интеллектуальный субъект, программа, система должны настроить и переориентировать свое поведение, если среда изменяется. При этом нет необходимости обращаться ко всей базе знаний. База должна настраиваться на определенную среду, ограничиваясь знаниями, которые к ней относятся. Получаем ориентированную базу знаний. А доступ осуществляется в зависимости от ситуации, в которой оказались субъект, программа, система.

В реальной жизни поведение определяется установленными правилами, которые связаны в определенный алгоритм поведения, связываемый с ситуацией. Эта регламентация отражает заданную структуру среды. Примером могут служить армейские уставы, которые предопределяют действия военнослужащих. Если ситуация отличается от установленных рамок, то возникает задача – определить свои действия, которые не предусмотрены регламентом. И здесь необходим интеллект, позволяющий связать новую ситуацию с возможными действиями интеллектуального субъекта.

Интеллектуализации программ

Продуктом программирования являются *программы*, в виде взаимосвязанной системы алгоритмов, воспринимаемых компьютером, и организованные в одну или несколько структур данные, обрабатываемые этой программой [12]. Роль структурированных данных в программировании огромна. Часто создание соответствующей структуры данных это существенная часть общего процесса программирования. Эта роль данных при программировании хорошо представлена в объектно-ориентированном программировании.

Обычно программа строится таким образом, чтобы она могла обрабатывать различные данные. Например, программа для решения систем линейных уравнений решает различные системы, заданные соответствующей матрицей. Сложность таких вычислений часто связывается с большой размерностью матриц. Чаще всего для таких программ заданы алгоритмы обра-

ботки и сами данные, а необходимость использования компьютера связывается с большим объемом вычислений в ходе применения алгоритмов обработки к данным.

Следующий класс программ определяется необходимостью интерфейса человека и компьютера в процессе обработки данных. В ходе интерфейса определяются зависящие от ситуации данные и ход применения алгоритмов обработки. В таких программах человек частично управляет процессом решения задач, используя свой интеллект, чтобы скорректировать процесс вычислений.

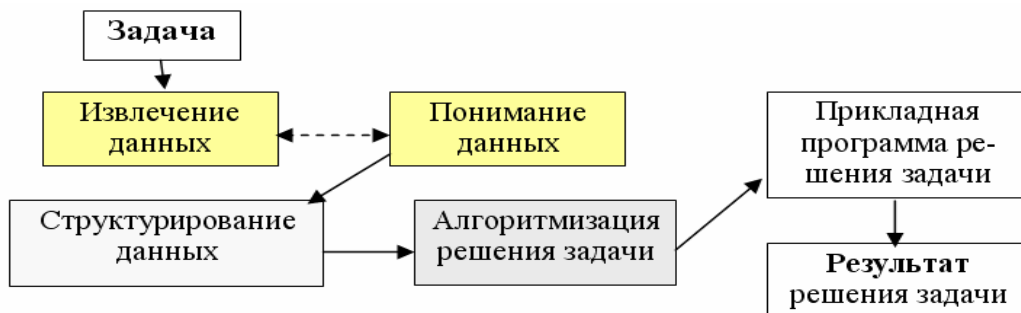
При этом необходимы программы, которые могли бы самостоятельно понимать возникающую ситуацию и настраиваться на различные задачи, находить их решение так, как это делает человек. Класс таких программ назовем *интеллектуальными программами*. Это своего рода метапрограммы, функцией которых является создание прикладной программы, применимой к тем условиям и данным, которые поступают в программу. Анализ этих данных позволяет интеллектуальной программе выбрать из базы своих знаний схему решения задачи, на основании которой затем строится исполняемая на компьютере программа. В настоящее время также используется близкое понятие интеллектуальных агентов [1, 10, 13].

Интеллектуальные программы (ИП) сначала выделяют и анализируют данные, входящие в поставленную задачу. С задачей обязательно связана внешняя среда, в которой эта задача рассматривается. Информация о среде либо вводится в ИП, либо самостоятельно выделяется ИП в процессе ее взаимодействия с базой знаний

как часть тех данных, которые анализируются программой. Затем выбирают или строят алгоритм, который мог бы обработать эти данные так, чтобы решить задачу. Этот алгоритм преобразуется в программу, которая уже выполняется на компьютере и дает ожидаемый результат. Общая схема такого процесса представлена на рисунке.

Формирование данных вместе с их извлечением из условия задачи, что часто является нетривиальным процессом, предполагает и понимание данных (на рисунке отмечено двусторонней стрелкой). И здесь необходим интеллект человека, поскольку от понимания данных зависит весь процесс решения задачи. Понять данные в их взаимоотношении и взаимосвязи друг с другом – это значит сделать существенный шаг к их представлению и структуризации, а затем и к погружению в среду, в которой решается задача, связав задачу со средой переходим к выбору аппарата, применимого для решения, и алгоритмизации самого процесса решения задачи.

У человека такие подпроцессы общего процесса решения опираются на его знания, опыт, имеющуюся у него информацию и методологию, т. е. на то, чем должен обладать субъект, решающий задачу. Например, предшественники теории относительности не смогли ее открыть. Э. Мах, формулируя известный принцип Маха, сознавал неполноту законов Ньютона. Но он не обладал необходимым математическим аппаратом, чтобы построить соответствующую теорию. А. Пуанкаре и Х. Лоренц имели такой аппарат, но не ввели его в физику, хотя А. Пуанкаре сформулировал принцип относительности за 10 лет до Эйнштейна. И только А. Эйнштейн акцентировал внимание физиков на принципе



рисунок

относительности и его физических следствиях, создав теорию относительности.

Для того, чтобы построить интеллектуальную программу необходимо, чтобы такая программа могла обращаться к опыту, которым обладает человек, и учитывать этот опыт в своей работе. По предположению, опыт представляется в виде базы знаний, которую задействует ИП в ходе своей работы. Часть алгоритмов, реализующих интеллект, являются составляющей ИП. Другая часть этих алгоритмов может находиться в базе знаний вместе с той информацией, которая используется при решении задачи. Выбирая алгоритмы и структурированные данные, ИП строит пучок возможных решений, из которого затем выбирает одно или несколько, передавая их пользователю – постановщику задачи. В любой момент работы ИП возможен возврат к предыдущему этапу и его повторение с учетом полученных результатов.

При работе современных программ в среде Windows, операционная система обменивается с программой сообщениями, которые являются реализацией взаимодействия со средой, в которой действует программа. Поэтому можно предположить, что так же, как любая современная программа работает в среде, создаваемой операционной системой, так и новый класс интеллектуальных программ должен, выполняясь на компьютере, погружаться в интеллектуальную среду, уже построенную в виде *системы интеллектуального представления*. Такой системой будет обеспечиваться каждый компьютер по аналогии с операционной системой. Оснащение компьютеров такими системами, наряду с операционной, определит будущий *четвертый революционный шаг* развития информатики – переход к постоянному использованию интеллектуальных программ.

Причем ИП наряду с самим решением сможет выдавать и комментарии к такому решению. Или варианты решения в различном виде. Выполнение действий ИП предполагает ее взаимодействие с пользователем, как при настройке программы, так и в ходе выполнения ею действий. На-

пример, решая квадратное уравнение, ИП может запросить, в каком виде выдавать решение – в виде выражения, включающего корни и мнимые числа, или в виде числовых выражений, имеющих приближенный вид, или в графической форме показывающей, как расположены корни на оси или плоскости.

Зато база знаний, к которой прикрепляется ИП, содержит в себе всю информацию о действиях для каждого варианта решения и структуру, определяющую, как строить соответствующий поставленной задаче алгоритм в заданной внешней среде. Следовательно, процесс алгоритмизации и программирования во многом определен той структурой, которая связывается с решаемой задачей.

Например, решение любой математической задачи всегда связано со средой, определяемой используемыми понятиями и представлениями. Скажем, доказательство бесконечности числа простых чисел предполагает, что такие понятия внешней среды, как натуральный ряд, простое число, операции над числами – сложение с единицей, умножения, деление – известны субъекту. Без этих знаний известное доказательство Эвклида окажется бессмысленным рассуждением. Именно созданию среды, в которой выполняются научные рассуждения, посвящена каждая наука, определяющая свой дискурс в виде взаимосвязанной системы понятий и представлений.

По отношению к внешним средам возможна следующая классификация ИП. Самый нижний – нулевой уровень – это наиболее распространенный уровень программ, в которых среда задана структурой данных, определенных самой программой. На этом уровне в настоящее время строится большинство программ. Для выполнения программы в нее вводятся данные, а программа реализует заложенный в нее единственный алгоритм до тех пор, пока не будет получено решение. Эти программы не обладают собственным интеллектом в силу предопределенности своих действий, определяемых заложенной в программу единственной структуризацией среды.

Более высоким является уровень программ, которые могут самостоятельно выбирать стратегию или алгоритм решения задачи, опираясь на существующую у них базу знаний. Формулируя задачу, пользователь заранее не знает, каким образом, каким алгоритмом эта задача будет решена. Анализируя данные, программа сама выбирает тот или другой путь, определяющий решение задачи, среди существующей в базе знаний вариантов структуризации среды.

Как обобщение этого уровня можно рассматривать вариант, при котором программа, исходя из полученного задания, сама выбирает стратегию, или конструирует алгоритм, используя заготовки из базы знаний и перебирая различные варианты возможных действий, объединенных с оценкой полученных после выполнения этих действий результатов. Такие оценки определяют приближение этих результатов к ожидаемому решению. В зависимости от их возможностей, которые имеют алгоритмы оценок, можно говорить о том, что здесь возникает такое понятие, как *смысл* полученных результатов, а сами программы обладают интеллектом.

Таким образом, обязательным свойством ИП является ее эффективное взаимодействие с базой знаний (БЗ). В этой базе представлена и структурирована внешняя среда, в которой работает ИП, решая поставленную задачу.

Первый шаг. В соответствии со схемой, представленной на рисунке, ИП сначала извлекает исходные данные и поставленную цель из задачи, которую нужно решить. При этом ИП опирается на БЗ, стараясь найти соответствие между задачей и знаниями в БЗ, чтобы связать задачу со структурой внешней среды.

Второй шаг. Понимание данных – это погружение извлеченных данных в выбранную область БЗ, сопоставление данных с представленной в БЗ средой. Здесь формируется *смысл задачи*, как интерпретация данных в выбранной среде. По предположению данные содержат и задание, и условия, определяющие ожидаемую цель.

Третий шаг. Обычно решение задачи определяется как путь, по которому в структуре, связываемой со структурной моделью среды, можно перейти от исходных данных к цели, определяющей результат решения задачи. Этот путь или пути, если их несколько, состоят из элементов структуры среды и должны быть построены ИП.

Четвертый шаг. Если каждому элементу найденного пути сопоставить оператор, то получаем алгоритм решения задачи. Если же операторы имеют разный вид, то получаются и разные алгоритмы. *Стратегия* выбора алгоритма задается правилами и условиями выбора пути решения задачи и операторов, сопоставляемых отрезкам пути.

Пятый шаг. Это компьютерное представление данных, связанных с задачей, и запись построенного алгоритма в виде исполняемой на компьютере программы, которая реализует все действия, представленные в виде операторов, по обработке данных и преобразованию их в результат.

Интеллект программы проявляется в том, что, опираясь на БЗ, программе самостоятельно удается найти структуру решения поставленной задачи в виде пути, определяющем переход от исходных данных к ожидаемой цели, и связать этот путь с алгоритмом решения, чтобы выполнить на компьютере необходимые действия в соответствии с этим алгоритмом.

Математическая модель

Для того, чтобы представить в математической форме схему работы ИП, сначала зададим формальное представление внешней среды, которую обозначим символом \mathcal{E} . По предположению \mathcal{E} состоит из объектов, имеющих различную структуру, и связей между ними. Элементарными объектами среды \mathcal{E} называются воспринимаемые в среде сущности (предметы, события, явления, факты), неделимые на части в рамках данной модели, составляющие среду атомы. Данные сущности

имеют имена и обозначаются символами a, b, c, \dots , которые могут иметь индексы, различающие сущности в множестве объектов среды \mathcal{E} . Свойства, присущие объектам, будем называть их признаками.

Между элементарными объектами в среде \mathcal{E} существуют связи, задаваемые в виде отношений, отображений, преобразований, выводов. Возможны два варианта. В первом, – связанные элементарные объекты, а в общем случае объекты вообще, образуют новую сущность, которая рассматривается как новый объект среды \mathcal{E} . Такой объект получает свое имя и рассматривается в дальнейшем как компонент среды. Это объект, имеющий внутреннюю структуру в среде. Кроме того, каждый элементарный объект a может рассматриваться как объект, если предположить существование связи 1_a , отображающей объект a сам в себя. Такую связь будем называть тождественной связью.

Во втором – связанные объекты, хотя и рассматриваются как объектные структуры внешней среды, но при этом их составляющие сохраняют свои имена и признак. Такие структуры отражают сложность и взаимные связи между объектами внешней среды. Но эти структурные формирования не рассматриваются как единая сущность, что было в первом варианте. Совокупность всех объектов внешней среды обозначим $Ob \mathcal{E}$.

Чтобы формализовать образование структур во внешней среде \mathcal{E} , допустим, что существует операция \oplus , определяющая формирование структур из объектов среды. Эта операция определена на конечных подмножествах, составленных из объектов среды \mathcal{E} , и сопоставляет конечному множеству объектов некоторую построенную из них структуру, которая является объектом \mathcal{E} . Операция $\oplus: [Ob \mathcal{E}]^* \rightarrow Ob \mathcal{E}$, $[Ob \mathcal{E}]^* = \{(a_i)_{i \in I} \mid a_i \in Ob \mathcal{E}\}$, а I – счетное множество, связанное с операцией \oplus , и включающее только конечные наборы произвольной длины. Если $I = \{1\}$, то $\oplus_{i \in I} x_i = x_1$.

В реальных средах обычно существует много таких операций структурирования. Но для формулировки простой модели, предлагаемой в настоящей работе, предположим существование одной такой операции. Среда \mathcal{E} называется N -структурированной, если операция \oplus определена только на конечных наборах объектов из $Ob \mathcal{E}$, причем длина таких наборов не превосходит числа N .

Между объектами среды имеются связи, определяющие взаимозависимость объектов между собой, влияние одних объектов на другие. Связи – это некоторые направленные отношения, существующие в среде и характеризующие воздействие одних объектов на другие. Связи обозначаются символами $\varphi, \psi, \theta, \dots$, которые могут быть различных видов и типов, представляться разными способами и играть различную роль.

Связи между объектами могут задаваться отношениями:

- следования, порядка, включения, сравнения, принадлежности;
- зависимости, совместности, логической связи;
- управления, подчинения;
- взаимодействия, преобразования (операции), влияния [11].

Связи могут задаваться функциями или операторами, которые показывают как выражается один объект через другой или другие, или как связаны одни объекты с другими. В реальных условиях не всегда известны все связи. В ходе решения задач, в процессе формирования поведения субъекта могут формироваться и новые связи, как и новые объекты, необходимые для построения адекватного поведения. Здесь и появляется та самая структуризация, которая связана с интеллектом.

Моделью среды \mathcal{E} будем называть формальную конструкцию α , состоящую из множества объектов $Ob \mathcal{E}$, и множества отображений между объектами, ассоциируемых со связями среды \mathcal{E} . Для этих отображений определена операция композиции, обозначаемая как $*$. Пусть заданы связи между объектами – $\varphi: a \rightarrow b$,

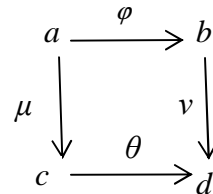
$\psi : b \rightarrow c, \theta : c \rightarrow d$. Будем говорить, что связи φ, ψ – совместимы, если для них определена композиция, результат которой совпадает со связью $\xi : a \rightarrow c$, $\varphi * \psi = \xi$. По определению тождественная связь 1_a совместима с любой связью вида φ , и композиция $1_a * \varphi = \varphi = \varphi * 1_b$. Если связи не совместимы, то по определению их композиция задает нулевое отображение (связь), т.е. $\varphi * \psi = 0_{ac}$. Композиция нулевого отображения с любым другим всегда дает нулевое отображение (связь). Будем считать, что композиция совместимых связей ассоциативна, $(\varphi * \psi) * \theta = \varphi * (\psi * \theta)$. Тогда конструкция \mathbf{a} , включающая объекты и связи между объектами, является категорией, связываемой со средой \mathcal{E} . Множество объектов \mathbf{a} совпадает с множеством $Ob \mathcal{E}$, а ее множество морфизмов определено связями между объектами.

Категория \mathbf{a} может быть расширена за счет введения в нее конечных копроизведений. В дальнейшем будем предполагать, что категория \mathbf{a} замкнута относительно таких копроизведений.

Задача в среде \mathcal{E} задается как множеством объектов $(a_1, a_2, \dots, a_n; z)$, где объекты a_1, a_2, \dots, a_n – это условия задачи, а объект z – ожидаемый результат и требование найти существующую связь или доказать отсутствие ее между условиями задачи и ее результатом. В категории \mathbf{a} решение задачи состоит в указании ненулевого морфизма, ведущего от копроизведения условий задачи $p = \coprod_i a_i$ к ее результату, представленному объектом z , или в доказательстве того, что такого морфизма в \mathbf{a} нет.

Иногда заранее известно, что решение задачи существует, но вопрос заключается в том, чтобы представить это решение в виде последовательности объектов, которые получаются при применении морфизмов, композиция которых определяет переход от объекта p к объекту z .

Введем две новые категории. Первая – категория связей \mathcal{C} (в [14] называется категорией стрелок). Ее объекты – это морфизмы категории \mathbf{a} . Если $\varphi : a \rightarrow b, \theta : c \rightarrow d$, то морфизм из φ в θ – это пара морфизмов $\langle \mu, \nu \rangle$, где $\mu : a \rightarrow c, \nu : b \rightarrow d$, для которых коммутативна диаграмма

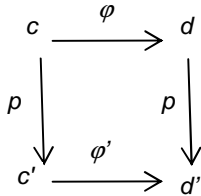


Вторая – это категория L -структур [11]. Определим **язык** L как множество конечных последовательностей объектов из \mathbf{a} , вида $A = a_1 a_2 \dots a_n$, называемых словом и удовлетворяющим условию: для любого i и пары объектов (a_i, a_{i+1}) , входящих в слово A существует в \mathcal{C} такая операция ω , что $\omega(a_i) = a_{i+1}$. Подъязыком $L(\theta)$ языка L , определенным множеством операций $\theta \subseteq \mathcal{C}$, называется множество слов языка L , использующим для своего формирования только операции из множества θ .

Структурным элементом $S(a, b)$ называется слово языка L , которое начинается объектом a и заканчивается объектом b . Соответственно, $S(a, b, \theta)$ – это структурный элемент, являющийся словом подъязыка $L(\theta)$. L – структурой $L_{a,b}^\theta = (a, b, \theta)$, определенной объектами a и b и множеством операций θ называется множество всех структурных элементов вида $S(a, b, \theta)$, использующих операции из θ . $L_{a,b}^\theta = \{S(a, b, \theta) = (x_1, \dots, x_n) \mid n \geq 2\}$. Каждый набор (x_1, \dots, x_n) , входящий в L – структуру $L_{a,b}^\theta$, назовем θ путем из a в b .

Морфизмом α L -структур $L_{a,b}^\theta$ в $L_{a',b'}^\theta$ называется пара отображений $\alpha = (p, q)$, где p – отображение множеств

ва α в α , $p: a \rightarrow a', p: b \rightarrow b'$, а q – отображение $q: \theta \rightarrow \theta'$, причем для любой пары объектов (c, d) , входящих в θ – путь из $L_{a,b}^\theta$, коммутативна следующая диаграмма:



где $\varphi' = q(\varphi)$. В силу совместимости \mathcal{C} операции из θ ассоциативны и существует тождественный морфизм L -структур. Поэтому множество L -структур вместе с вышеопределенными морфизмами образует категорию L -структур, которую обозначим S .

Определим функтор P из категории связей \mathcal{C} в категорию S . Каждому объекту $\varphi: a \rightarrow b$ из множества $Ob \mathcal{C}$ функтор P сопоставляет L - структуру: $P(\varphi) = L_{a,b}$, а морфизму соответственно морфизм L -структур. Этот функтор назовем программным функтором, если рассматривать морфизм $\varphi: a \rightarrow b$ как задачу, а L -структуру как множество ее решений.

Заключение

Наполнение интеллектом систем, программных комплексов, техники, использующих для обеспечения своих действий компьютеры и базы знаний, является одним из основных направлений развития современного программирования. В определенной степени исчезли технические препятствия, не позволявшие в реальном времени работать с большими объемами информации. Это позволяет компьютерам оперировать с поступающими данными, не упрощая их. А человеку передать компьютеру часть работы по составлению программ, решающих задачи в изменяющихся

внешних средах. Интеллектуальная программа строит свои действия на основании заложенного в нее опыта, что позволяет существенно расширить ее возможности.

С другой стороны, безграничной оказывается область применения интеллектуальных систем, что предполагает изменение самого общества. Последнее десятилетие все настойчивее говорят о наступающей цивилизации знаний. А такая цивилизация не только будет основываться на интеллектуальных системах, но она невозможна без них. Поэтому таким важным представляется продвижение по пути разработки интеллектуальных систем, технологией создания которых является программирование.

1. Капитонова Ю.В., Летишевский А.А. Парадигмы и идеи академика В.М.Глушкова. – Киев: Наук. думка, 2003. – 454 с.
2. Letichevsky A.A., Kapitonova Y.V., Konozenko S.V. Computation in APS // Theor. Comput. Sci. – 1993. – 119. – P. 145 – 171.
3. Андон Ф.И., Дорошенко А.Е., Цейтлин Г.Е., Яценко Е.А. Алгеброалгоритмические модели и методы параллельного программирования. – Киев: Академперіодика, 2007. – 634 с.
4. Поспелов Д.А. Ситуационное управление: теория и практика. – М.: Наука, 1986. – 288 с.
5. Мейтус В.Ю., Мейтус В.В. Массы, движения, революции. – Киев: Ника-Центр, 2008. – 500 с.
6. Хокинс Д., Блейкли С. Об интеллекте. – М.: Вильямс, 2007. – 240 с.
7. Глибовець М.М., Олецький О.В. Штучний інтелект. – К.: КМ Академія, 2002. – 366 с.
8. Рассел С., Норвиг П. Искусственный интеллект. Современный подход. – Пер. с англ. – М.: Вильямс, 2006. – 1408 с.
9. Gottfredson L.S. Mainstream Science on Intelligence // Wall Street J. December 13, 1994. – P. A18.
10. Мейтус В.Ю. К проблеме интеллектуализации компьютерных систем // Математичні машини і системи. – 2008. – № 2. – С. 24 – 37.

11. Мейтус В.Ю. К проблеме интеллектуализации систем управления // Матеріали XIII Міжнар. конф. з автоматичного управління (Автоматика – 2006). Вінниця, 25-28 вересня 2006 р. – Вінниця, 2006. – С. 466 – 471.
12. Вирт Н. Алгоритмы + Структуры данных = Программы. – М.: Мир, 1985. – 406 с.
13. Wooldridge M., Jennings N. Intelligent Agents: Theory and Practice // The Knowledge Engineering Review. – 1995. – Vol. 10(2). – P. 115 – 152.
14. Голдблатт Р. Топосы. Категорный анализ логики. – М.: Мир, 1983. – 488 с.

Получено 18.09.2008

Об авторе:

Мейтус Владимир Юльевич,
доктор физико-математических наук,
ведущий научный сотрудник.

Место работы автора:

Международный научно-учебный
центр информационных технологий и
систем НАН и МОН Украины,
03680 Киев-187,
проспект Академика Глушкова, 40.
Тел.: (044) 526 1319
e-mail: vmeitus@gmail.com.