

тельств над великим математиком и его геометрией... Признание пришло к великому математику лишь после его смерти... Как ненаучные рассматривались вначале результаты исследований А.Л. Чижевского, создателя гелиобиологии – науки о воздействии космических процессов, в первую очередь солнечной активности, на массовые явления и процессы, которые протекают в земной биосфере, человеческом организме и обществе» [Берков. 2004. С. 42-43].

Литература

1. Берков В.Ф. Философия и методология науки. М., 2004.
2. Введенский А.И. Логика как часть теории познания. Третье издание. Пг., 1917.
3. Войшвилло Е.К., Дегтярёв М.Г. Логика как часть теории познания. Книги I, II. М., 1994.
4. Ивлев Ю.В. Теория и практика аргументации. М., 2008.
5. Мулдашев Э. Р. От кого мы произошли. М., 2002.

Трансформация видов компьютерной репрезентации информации в эволюции алгоритмических языков программирования

к.ф.н. доц. Иноземцев В.А.
Университет машиностроения
inozem_63@mail.ru

Аннотация. В статье проводится исследование перехода от репрезентации информации в виде компьютерных данных к репрезентации информации в виде компьютерных знаний в ходе эволюции алгоритмических языков программирования.

Ключевые слова: компьютерные данные, компьютерные знания, компьютерная программа, языки программирования, алгоритмические языки, эволюция алгоритмических языков программирования, процедурный тип компьютерной репрезентации знания.

Исследование выполнено при финансовой поддержке РГНФ в рамках проекта проведения научных исследований («Логический инструментарий и философские основания современной науки»), проект № 14-23-01005.

Рассмотрим в работе каким образом в процессе эволюции алгоритмических языков программирования происходил переход от репрезентации информации в виде компьютерных данных к репрезентации информации в виде компьютерных знаний. Для этого вначале обратимся к таким терминам как «компьютерные данные» и «компьютерные знания».

Под компьютерными данными (или просто данными) в информатике понимают совокупности конкретной числовой и вербальной информации об объектах некоторых фрагментов действительности, называемых предметными областями, и присущих этим объектам свойствах. Информация, заключённая в компьютерных данных, репрезентируется в виде формализованных документов (файлов, таблиц, списков и т.д.) [3, 4].

Компьютерными (машинными) знаниями в информатике и теории искусственного интеллекта называют вводимую с помощью специальных процедур в базы и банки знаний интеллектуальных компьютерных систем информацию о закономерностях структуры и функционирования предметных областей действительности [3, 4]. Для репрезентации такого рода информации используются совокупности высказываний, формализуемых с помощью логических исчислений (исчисления высказываний и исчисления предикатов). Кроме того, для этих целей применяют различного рода сетевые и фреймовые структуры, вошедшие в состав исследований в области искусственного интеллекта под влиянием работ в когнитивной психологии, которые базируются на концепциях структур человеческого восприятия и памяти, разработанных в этом разделе психологической науки.

Термин «компьютерные данные» появляется в конце 50-х годов XX века в информатике, где он по преимуществу с тех пор и используется. Термин «компьютерные знания» вхо-

дит в научный лексикон в конце 1960-х – начале 1970-х годов в связи с расцветом исследований в области искусственного интеллекта (ИИ). Именно в ИИ этот термин преимущественно и применяется и именно он наиболее тесно связан с логикой. Компьютерные данные и компьютерные знания являются достаточно близкими видами репрезентации информации, и между ними не существует жесткой демаркации. Компьютерные знания «вырастают» из компьютерных данных и являются более высоким этапом их развития.

Рассмотрим далее эволюцию алгоритмических языков программирования, ставшую одним из важнейших факторов трансформации видов репрезентации информации от компьютерных данных к компьютерным знаниям. Для этого обратимся вначале к анализу основных этапов развития программирования и сопутствовавшей этому процессу эволюции языков программирования. Разработка и использование средств электронно-вычислительной техники оказываются теснейшим образом связанными с компьютерными программами и компьютерными данными. Существуют различные трактовки понятия «компьютерная программа». Объединяющим для всех трактовок является то, что компьютерные программы представляют собой непустые последовательности правил и предписаний, которые распознаются программируемыми объектами и реализуются ими в виде последовательностей операций.

На первом этапе развития программирования в 50-е годы XX века, когда компьютеры применяются как большие арифмометры для решения различного рода вычислительных задач, доминирующую роль играют реализуемые на компьютерах программы, в которых овеществляется процесс решения задач. Роль данных на этом этапе является менее значительной. В этот период с программами оказываются тесно связанными все компьютерные данные, и без программ они не могут существовать. На этом этапе программирования компьютеры выступают как формальные системы, на вход которых поступают программы, написанные на языках машинных кодов, а на выходе получают некоторые стандартизированные записи. Такой взгляд на компьютеры впервые высказывает Дж. фон Нейман. Он доказывает, что функционирование компьютеров можно уподобить работе машины Тьюринга, точнее так называемой Линейно ограниченной машине, в силу того, что память компьютеров не является бесконечной.

Второй этап развития программирования, очерченный временными рамками 60-х – 70-х годов XX века, характеризуется тем, что компьютеры начинают использоваться как универсальные машины для накопления и обработки всех видов информации. На этом этапе возрастает роль данных, данные отделяются от программ, появляются базы данных различной структуры. Компьютерная репрезентация и обработка данных на этом этапе выявляют их особый эпистемологический статус. Первоначально компьютерные данные представляют собой результат фиксации единичных наблюдений, измерений, экспериментов, фактов или ситуаций. При этом информационная ценность таких компьютерных данных постоянно меняется. На основе разрозненных компьютерных данных невозможно получить никаких прагматических предписаний и правил, которые могут регулировать человеческую деятельность. Формулирование таких правил является характерным признаком более сложного вида компьютерной репрезентации информации – компьютерных знаний. Для того чтобы компьютерные данные стали применяться людьми, их необходимо инкорпорировать в контекст компьютерных знаний, определенным образом сопоставить с ними, и после этого эффективно использовать для получения новой информации.

На втором этапе развития программирования в 1960-е – 1970-е годы происходит серьёзный переворот в идеологии, методологии и технологии программирования. Оно начинает осуществляться в этот период автоматически посредством комбинирования фрагментов программ или подпрограмм из алгоритмов или отдельных программных блоков, записанных в памяти компьютеров и конструируемых согласно специальным предписаниям. Первостепенное значение при этом приобретают такие этапы программирования, как анализ и формализация задач, системная разработка программ и блоков программ, анализ и устранение ошибок, разработка программного инструментария и ряд других. Именно на этом этапе развития

программирования происходит формирование из компьютерных данных компьютерных знаний и оформление декларативного и процедурного типов компьютерной репрезентации знания. В этот период появляются базы знаний [2] и оформляются логическая, сетевая и фреймовая концепции компьютерной репрезентации знания.

Наконец, третий этап развития программирования наступает в 1980-е – 1990-е годы и продолжается до настоящего времени. Его формирование вызывается появлением персональных компьютеров и дальнейшими трансформациями в технологии и инструментарии программирования. Это период автоформализации, связанный с простотой, удобством в обращении и массовостью компьютеров. Происходит переход от решения отдельных задач к программам, способным в автоматическом режиме решать серии взаимосвязанных задач. Процесс же формирования из компьютерных данных компьютерных знаний в основном завершается ещё в 1970-х годах на втором этапе развития программирования.

Развитие программирования детерминирует эволюцию языков программирования. Переход к новому информационному обществу, происходящий в последние десятилетия, вызывает серию революционных трансформаций в разработке и использовании языков программирования.

Выделим следующие этапы в развитии языков программирования: 1) этап языков машинных кодов (1950-е годы); 2) этап машинно-ориентированных символьных языков программирования (конец 1950-х – начало 1960-х годов); 3) этап проблемно-ориентированных или алгоритмических языков программирования (середина 1960-х – 1970-е годы); 4) этап современных языков программирования (объектно-ориентированные языки, дескриптивные языки). Рассмотрим специфику и сферы приложения каждого из типов языков программирования. Как известно, компьютерные программы записываются на определенных языках программирования. Все языки, в том числе языки программирования, представляют собой знаковые системы, обладающие рядом функций (иногда частично пересекающихся): репрезентации внеязыковой реальности, хранения и передачи информации, закрепления смыслов и значений языковых выражений. Эти функции являются необходимыми для знаковых систем, которые применяются в качестве языков.

Первыми языками программирования становятся языки машинных кодов. Они, как простейшие из языков программирования, которые понимают компьютеры, представляют собой языки двоичных кодов, то есть системы последовательностей двузначных цифр, построенных по определенным правилам. Создание программ на языках машинных кодов вызывает необходимость в том, чтобы программисты на этом раннем этапе развития языков программирования могли бы точно описывать посредством двузначных последовательностей все процедуры, операции, данные и рабочие устройства компьютеров. Такая деятельность, характерная для программистов 50-х годов XX века, требует значительного количества времени, является чрезвычайно дорогостоящей, нуждается в выполнении высококвалифицированными специалистами и не позволяет создавать сложные программы. Программирование на языках машинных кодов представляет собой очень трудоёмкий процесс. Программистам при работе с языками машинных кодов необходимо не только выписывать все команды, определять их последовательности, но также точно и строго организовывать память, хранить все полученные результаты, в том числе промежуточные. При программировании на языках машинных кодов программы получаются ненаглядными. Кроме того, для реализации одного и того же алгоритма на разных компьютерах необходимо писать разные программы в силу того, что компьютеры имеют на этом этапе собственные системы команд. Это подчас приводит к дублированию работы в случае решения одной и той же задачи на разных компьютерах. Содержательный смысл данных, которые на этом этапе вводятся в компьютеры и выводятся из них как результаты, оказывается скрытым от программистов. Смысловая сторона данных может интерпретироваться только теми программистами, которые работают в определённых предметных областях.

Исходные данные, с которыми работают программы, организуются на первом этапе развития языков программирования по однозначному, жёсткому принципу. Данные пред-

ставляют с точки зрения компьютеров машинные слова фиксированной длины. Как правило, это длины элементов компьютерной памяти – машинных ячеек. В машинных словах хранится определённая информация (числа, команды, последовательности символов и т.д.). Массивы данных также не структурируются на первом этапе развития языков программирования. Каждый элемент из массива данных или совершенно не зависит от других элементов, или зависит только «пространственно». Во втором случае при отображении значения элементов массива данных в памяти компьютеров приходится располагать пространственно связанные элементы данных в виде адресно-связанных ячеек, то есть этим элементам необходимо следовать друг за другом в соответствии с нумерацией машинных слов в памяти компьютеров.

Языки программирования второго поколения, появившиеся в конце 1950-х годов, являются машинно-ориентированными языками, так как эти языки ориентируются на машинные команды, которые исполняются компьютерами. При использовании машинно-ориентированных языков становится возможным разделить организацию памяти компьютеров и написание сложных программ. Языки второго поколения являются языками символического программирования. В них создаются дополнительные программные средства, которые управляют отдельными блоками вычислительных операций. Происходит встраивание таких блоков в основные программы или использование их как средств поддержки программ. Машинно-ориентированные языки, будучи символическими языками программирования, упрощают процесс программирования и обеспечивают доступ к программированию значительному количеству пользователей.

Языки программирования третьего поколения – проблемно-ориентированные или алгоритмические языки программирования – появляются в середине 1960-х годов и интенсивно развиваются в 1970-е годы. Эти языки ориентируются, в первую очередь, на точные формулировки проблем. Перевод программ, написанных на языках второго и третьего поколения, на языки машинных кодов проводится посредством особого класса программ, так называемых трансляторов. Трансляторы представляют собой специальные средства для перевода (трансляции) записей программ, написанных на языках второго и третьего поколения, в записи тех же программ на языках машинных кодов. Необходимость в таком переводе возникает в связи с тем, что компьютеры могут понять программы, зафиксированные на машинно-ориентированных и проблемно-ориентированных языках лишь в том случае, если они соответствующим образом автоматически, с помощью трансляторов, переводятся на языки машинных кодов. Проблемно-ориентированные языки дают импульс языкам программирования для ИИ [5].

Развитие алгоритмических языков программирования в 1960-е – 1970-е годы осуществляется по двум основным направлениям. Первая линия эволюции алгоритмических языков программирования в это время приводит к появлению процедурного типа компьютерной репрезентации знания. Примерами процедурного типа репрезентации знания в компьютерных науках являются знания, хранящиеся в пакетах прикладных программ [1, 3]. Вторая линия эволюции алгоритмических языков программирования завершается формированием декларативного типа компьютерной репрезентации знания. В качестве примеров декларативного типа выступают логические, сетевые и фреймвые модели и языки репрезентации знания, а также знания, хранящиеся в базах знаний [1, 2].

Первая линия эволюции алгоритмических языков программирования, которая завершается созданием процедурного типа компьютерной репрезентации знания, осуществляется посредством укрупнения команд, составляющих программы, в обобщённые операторы, представляющие собой управляющие конструкции (или предписания) алгоритмических языков программирования в каждой из предметных областей. В случае, если программисты работают с обобщёнными операторами, то, кроме ввода программ и данных в компьютеры, они должны также описать эти операторы и вводить их в уже упомянутые специальные программы трансляторов. В ходе выполнения программ трансляторы расшифровывают все необходимые операторы согласно введённым описаниям в соответствующие прикладные программы, которые постоянно хранятся в памяти компьютеров. Программы трансляторов и

прикладные программы отрываются от основных программ.

В том случае, когда требуется решать много задач из данной предметной области, трансляторы и прикладные программы остаются в памяти компьютеров на всё время решения основных задач. Если происходит смена предметных областей, то пакеты накопленных прикладных программ, использовавшиеся для определённой предметной области, называют именем этой области и переводят из оперативной памяти компьютеров в их основную память. Таким образом, в основной памяти компьютеров постепенно накапливается богатый набор прикладных программ из различных предметных областей. То обстоятельство, что эти программы хранятся в памяти компьютеров и тогда, когда они не нужны для решения очередных задач, превращает их в компьютерные знания. Компьютерные знания, хранящиеся в пакетах прикладных программ, как уже отмечалось ранее, относятся к процедурному типу компьютерной репрезентации знания. Программы трансляторов ориентируются исключительно на особенности языков программирования, используемых для решения задач, и не зависят от предметных областей, избранных пользователями. Их функционирование есть некоторые внутренние метапроцедуры. Таким образом, первая линия эволюции алгоритмических языков программирования приводит к появлению процедурного типа компьютерной репрезентации знания.

Почти одновременно с первой линией, хотя несколько позднее неё, происходит развитие второй линии алгоритмических языков программирования. Этот процесс начинается с того момента, когда компьютеры используют для решения не только вычислительных, но также общеинформационных задач. Если компьютеры оказываются включёнными в автоматизированные информационные системы, в их памяти постоянно хранятся определённые массивы данных. Входные программы представляет собой некоторые запросы на поиск информации из хранимых массивов. Эти программы называют поисковыми. Они ориентируются только на принятые в компьютерах способы репрезентации данных, и не зависят от избранных пользователями предметных областей, к которым относятся хранимые в памяти массивы данных. Это приводит к появлению структур, пригодных для хранения в памяти компьютеров информации о предметных областях.

Дальнейшая эволюция данного направления происходит по пути совершенствования способов репрезентации данных в памяти компьютеров. Последовательное развитие способов репрезентации данных от размещённых в ячейках памяти компьютеров машинных слов к файлам, спискам, таблицам, а от них к абстрактным типам данных приводит к качественному изменению данных. В результате развития способов репрезентации данных происходит переход от репрезентации информации в виде компьютерных данных к репрезентации информации в виде компьютерных знаний, а также переход от баз данных к базам знаний. Компьютерные знания, хранящиеся в базах знаний, относятся к декларативному типу компьютерной репрезентации знания. Поисковые программы также не зависят от предметных областей, избранных пользователями, и реализуют внутренние метапроцедуры. Вторая линия эволюции алгоритмических языков программирования приводит к формированию декларативного типа компьютерной репрезентации знания.

Таким образом, можно сделать вывод, что компьютерные знания появляются тогда, когда данные отрываются от обрабатывающих их программ, то есть у компьютерных данных начинает отсутствовать ориентация на конкретные программы. Это происходит в силу того, что данные отрываются от конкретных решаемых в данный момент на компьютерах задач. Однако эти данные могут вызываться для использования по желанию пользователей. Другим условием появления компьютерных знаний является разработка программ трансляторов и поисковых программ, которые не зависят от избранных пользователями предметных областей.

Следует отметить общие черты, присущие языкам программирования первых трех поколений. Когда программы записываются на этих языках, предполагается, что программисты достаточно подробно передают компьютерам, как нужно решать задачи, перечисляя все необходимые при этом действия, операции и их последовательности. В языках четвертого по-

коления (объектно-ориентированных и дескриптивных языках, появившихся в 1980-е годы) точно формулируются только решаемые задачи. Это становится возможным благодаря произошедшему в этот период скачку в развитии программирования. Программы, которые пишут на объектно-ориентированных и дескриптивных языках, определяют, что должны сделать компьютеры или что они должны решить. То, как это нужно сделать, компьютеры в этом случае определяют сами с помощью тех возможностей и тех программных модулей и блоков, которые у них имеются. Разработка подобного рода языков является существенным ускорением в направлении создания современных интеллектуальных систем.

Литература

1. Дейт К. Введение в системы баз данных. М., Наука. 1980. 463 с.
2. Мартэн Д. Базы данных. М., Радио и связь. 1983. 168 с.
3. Поспелов Д.А. Логико-лингвистические модели в управлении. М., Энергоатомиздат. 1981. 231 с.
4. Цаленко М.Ш. Моделирование семантики в базах данных. М., Наука. 1989. 287 с.
5. Эндрю А. Искусственный интеллект. М., Мир. 1985. 265 с.

ОСМЫСЛЕНИЕ ОБЩЕСТВА: ФИЛОСОФИЯ, МЕТОДОЛОГИЯ, ИССЛЕДОВАНИЯ

Необъективное и объективное сомнение в познании

Жданов С.Г.

Университет машиностроения
8(495) 683-99-70

Аннотация. Статья посвящена анализу процесса познания, выявляет роль, формы и виды сомнения. Рассматривается вопрос о месте интеллектуальной интуиции в процессе осознания проблемной ситуации.

Ключевые слова: интеллектуальная интуиция, объективное сомнение, необъективное сомнение.

Познавательная ситуация почти всегда включает наивное допущение о существовании изучаемого объекта в реальном мире, окружающем человека. Она как бы втягивает критически мыслящего человека “внутрь себя”, и вот уже наивные сомнения и вопросы о существовании объекта превращаются в глубинные вопросы, последовательность которых способна превратиться в развивающуюся систему, в чем собственно и проявляется свобода и щедрость познания.

Рассмотрение вопроса о динамике научного знания предполагает сравнение его постпозитивистских моделей. При этом важно не только исследовать историю знания как такового, но и выделить внешние и внутренние компоненты знания. Попперовская модель динамики научного познания, подчиняет развитие науки «внешнему» фактору – системе строгих стандартов. “Внешний фактор” свидетельствует, что мы стремимся к “внешне” строгому знанию и более строгому критическому незнанию о мире, о невозможности получить знание из опыта. Казалось бы, социальные и психологические причины научного поиска, естественно отнести к “внешним факторам”, однако у философов науки Т. Куна, М. Полани, П. Фейерабенда они больше соотносимы с “внутренней” динамикой науки. Для Т. Куна факты лишь относительно ценны, поскольку они включаются “во внешнюю” по отношению к фактам парадигму.

Исторически схожее разделение на внутреннюю и внешнюю динамику знания можно проследить уже в античности. Сократ ставит задачей своего познания “Любовь к Мудрости”. Он ищет такие особенные “внутренние” предикаты суждений, которые были сокрыты в глубинных тайнах души. Душа не сомневается “в себе” как субъекте, однако она стремится, возможно бессознательно, к новым признакам совершенства в своем преображении “в Люб-