



Проблемы логики и методологии науки

ПРОБЛЕМА ИЗВЛЕЧЕНИЯ ЗНАНИЙ В СИСТЕМЕ ВЗАИМОДЕЙСТВИЯ ЧЕЛОВЕКА И КОМПЬЮТЕРА: ОНТОЛОГИИ И ПРЕСУППОЗИЦИИ*

Д.Е. Пальчунов, В.В. Целищев

Данная работа посвящена проблемам извлечения, представления и передачи знаний. Рассматриваются два вида обмена знаниями: первый – между людьми, второй – между человеком и компьютером, т.е. между программной системой и ее пользователем. При этом необходимо отметить, что второй вид обмена знаниями во многом сводится к первому: проблемы непонимания, искажения или потери информации при общении пользователя с программной системой обусловлены различным восприятием мира, различными онтологиями и пресуппозициями, имеющимися у этого пользователя и у разработчиков данной программной системы. Соответственно, значительную важность приобретает проблема организации совместной деятельности человека и компьютера. Для решения этой проблемы необходимо решать задачи разработки человеко-машинных интерфейсов. Проблемы человеко-машинных интерфейсов тесно связаны с различием в представлении знаний у различных групп пользователей и разработчиков программного обеспечения. В этом контексте представляет интерес исследование факторов наличия или отсутствия единого понимания информации и знаний.

Ключевые слова: знание, информация, компьютер, онтология, человек

Проблемы восприятия знаний, обмена знаниями и передачи знаний. Обмен знаниями между человеком и компьютером

Наиболее отчетливо проблема экспликации понятия знания проявляется в процессе передачи знаний, и особенно новых знаний, что видно при анализе методов построения программных интерфейсов. Это обус-

* Исследования, результаты которых отражены в данной статье, поддержаны грантом Сибирского отделения РАН «Принципы построения онтологии на основе концептуализаций средствами логических дескриптивных языков», № 3.

ловлено тем, что в последние 20–30 лет возникла совершенно уникальная ситуация, которой ранее в мире не наблюдалось. Раньше знания, интересные, понятные и важные для широкой публики, появлялись достаточно медленно. После получения базового образования человек кроме узкопрофессиональных сведений мог узнавать новые маршруты путешествий, получать описания новых мест, событий, узнавать новые кулинарные рецепты и т.д. Мы говорим здесь именно об общих для всех, а не о профессиональных знаниях.

Характерно то, что, во-первых, у человека не было жесткой необходимости овладеть этими знаниями. Во-вторых, такие знания были специфичными: например, кулинарные рецепты и излагались в стандартных и привычных структурах – в виде кулинарной книги и т.п. В силу этого новыми знаниями человеку было легко овладеть, и не возникали такие проблемы, как правильность понимания полученных знаний.

Сейчас мы имеем принципиально иную ситуацию. Человек, который хочет смотреть фильмы на компьютере, записывать диски, писать письма при помощи текстовых редакторов и т.д., должен овладеть совершенно новыми и порой совершенно непривычными по структуре изложения знаниями – знаниями о том, как пользоваться данной программной системой и какие (совершенно новые и зачастую непривычные) возможности она предоставляет. Например, чтобы иметь возможность просматривать любые фильмы, которые можно скачать в Интернете, человек должен установить на своем компьютере достаточно большой набор кодеков. Для этого он должен узнать, что такое кодеки, какими они бывают, где их взять, как установить и т.д. То есть он должен овладеть достаточно большим набором новых весьма специфичных знаний. Если сравнить эти знания с теми, которыми люди овладевали 200 лет назад, то такие знания гораздо больше похожи на узкопрофессиональные, чем на общие для всех людей.

Новые знания стали массовыми: огромное число людей овладевают методами использования Microsoft Word и других программ, входящих в состав Microsoft Office. Объем знаний, необходимый для успешного использования Microsoft Word, является немалым и значительно превышает объем информации, содержащейся в наборе кулинарных рецептов, которыми пользуется тот же самый человек. Эти знания не только объемны, но и весьма разнородны и специфичны, – чтобы убедиться в этом, достаточно взглянуть на меню Microsoft Word 2007. С другой стороны, изложение таких знаний очень плохо структурировано. Интерфейсы большинства программных систем создаются их разработчиками по

принципу «кто во что горазд». Это является одной из главных проблем при обмене знаниями между человеком и программной системой.

Проблемы обмена знаниями и передачи знаний при разработке человеко-машинных интерфейсов

При разработке высококачественных человеко-машинных интерфейсов крайне важно обеспечить передачу знаний без потерь и искажений. Проблемы человеко-машинного интерфейса – это в первую очередь проблемы обмена знаниями между пользователями программного обеспечения и его разработчиками. Более того, это проблемы донесения знаний от разработчиков программного обеспечения до совершенно разнообразной аудитории, например аудитории всех пользователей Microsoft Office.

Ситуация осложняется тем, что, во-первых, обмен знаниями между разработчиками и пользователями программного обеспечения находится в рамках очень сжатого способа изложения – меню программной системы. Во-вторых, донесение знаний от разработчиков до пользователей происходит без обратной связи. Разработчики не имеют представления о следующих факторах:

- о степени понимания пользователем полученной информации (понятна она или непонятна);
- о том, как информации воспринята, что из нее и как именно пользователь понял;
- о том, каково мнение пользователя о полноте полученной информации, о его нерешенных проблемах и т.д.

Собственно, проблема отсутствия обратной связи при получении пользователем знаний о работе программной системы состоит в невозможности проверки, правильно ли пользователь понимает полученную информацию или нет. Даже если у пользователя есть ощущение полного понимания, это ощущение может оказаться совершенно ложным.

Таким образом, первая проблема, с которой мы сталкиваемся, – это проблема верификации правильного понимания получаемых сведений, отсутствия потери и искажения информации в процессе обмена знаниями. Данная проблема является достаточно общей, но учитывать ее также исключительно важно при разработке качественных человеко-машинных интерфейсов. То, насколько хорошо решены проблемы обмена знаниями, передачи знаний без потерь и без искажений, зачастую опре-

деляет успех или провал продвижения на рынок того или иного программного продукта.

Задача разработчиков программной системы – дать пользователю возможность удобного, понятного и быстрого обмена знаниями с программной системой. Здесь мы намеренно употребляем словосочетание «обмен знаниями», а не «обмен информацией». Именно имеющееся у разработчиков программной системы представление о пользовательском интерфейсе как об обмене информацией, а не обмене знаниями и является одной из главных причин возникающих проблем и плохого качества интерфейса. Например, для правильного понимания низкоуровневой информации, данных (таких, например, как температура, громкость, яркость) не нужны онтологии предметной области. Для понимания таких данных достаточно «верхнеуровневой онтологии», т.е. владения естественным языком в том объеме, в каком им владеет любой носитель данного языка.

С другой стороны, для правильного понимания меню достаточно сложной программной системы необходимо владеть весьма большим набором специальных терминов и понятий. От того, насколько это понимание совпадает у разработчиков программной системы и у пользователей, зависит качество разработанного интерфейса. Таким образом, необходимо как минимум рассматривать онтологии и presupпозиции для правильного понимания диалога между человеком и компьютером.

Поэтому общение между пользователем и программной системой правильно рассматривать не как обмен информацией, а как обмен знаниями. В определенном смысле программная система является здесь посредником между ее пользователями и разработчиками: она спрашивает у пользователя и сообщает ему ровно то, что считает нужным спросить или сообщить разработчик. Таким образом, сложности, проблемы и феномены, возникающие при обмене знаниями между пользователем и программной системой, практически те же, что и при обмене знаниями между людьми. Но кроме того возникают и новые, специфические сложности и проблемы [1].

Онтологии при обмене знаниями

Первая из этих проблем, которую мы сейчас рассмотрим, – это проблема различия онтологий у субъектов, обменивающихся знаниями. Как будет показано ниже, различие онтологий – один из главных моментов, обуславливающих непонимание или неправильное понимание между

субъектами, обменивающимися знаниями. Таким образом, использование понятия онтологии необходимо для обеспечения правильного обмена знаниями между пользователем и программной системой. В данной работе выдвигается тезис, что в распространенное определение онтологии входит понятие аналитического утверждения. Само определение может быть сформулировано так:

Формальной онтологией предметной области SD назовем пару $O = \langle A, \sigma \rangle$, где σ – множество ключевых понятий предметной области; A – множество аналитических предложений, описывающих смысл этих ключевых понятий.

При некоторого рода упрощающих предположениях любое логическое высказывание можно считать аналитическим. Тогда аналитические предложения, истинные в данной предметной области, определяют ее онтологию. Множество всех – аналитических и синтетических – предложений, истинных в данной предметной области, образует теорию этой предметной области. В таком контексте полезным будет следующее определение:

Множество T предложений, которые являются верными в каждом экземпляре предметной области SD , будем называть теорией предметной области SD .

Вообще, говоря об онтологии, надо говорить не об аналитических утверждениях, а о концептуальных истинах, которые определяются принятой концептуальной схемой, а не эмпирическими фактами. Памятуя, что онтология, согласно классическому определению, есть спецификация концептуализации, апелляция к концептуальному знанию здесь вполне уместна. Позднее мы покажем, что концептуальное знание играет ключевую роль и в понимании пресуппозиций.

Как мы показали выше, в ситуации обмена информацией и знаниями между пользователем и программной системой реально имеются три субъекта: пользователь, программная система и разработчики программной системы. Однако для пользователя разработчики скрыты «за программной системой», поэтому нам достаточно рассматривать только онтологию программной системы, на которую «спроецирована» онтология ее разработчиков. С другой стороны, пользователи программной системы нам интересны не только сами по себе, с их пониманиями, непониманиями, ожиданиями и пресуппозициями, – нам

также важны и наборы тех задач, которые пользователи решают с помощью данной программной системы. Поэтому кроме онтологии пользователя мы также будем рассматривать онтологию пользовательских задач.

Несложно понять, что онтология пользовательских задач и онтология программной системы существенно отличаются друг от друга. Прежде всего, рассмотрев интерфейсы и, соответственно, онтологии различных программных систем, которые решают данный класс задач, легко убедиться, что они имеют разные онтологии. Эти онтологии, конечно, пересекаются друг с другом, однако, во-первых, одни и те же термины в разных программных системах могут иметь разный смысл. Во-вторых, для представления знаний о функциональности и способах управления ею в разных программных системах используются разные наборы понятий. В качестве примера можно посмотреть, насколько различаются наборы понятий и их структура в интерфейсах MS Word 2003 и MS Word 2007. Кроме того, онтология пользовательских задач описывает в первую очередь цели, которые нужно достигнуть пользователю. Онтология программной системы предназначена для описания методов и средств достижения этих целей.

Таким образом, мы имеем две различные онтологии: онтологию пользовательских задач и онтологию данной программной системы. Для реализации взаимодействия пользователя с программой необходима интеграция онтологии пользовательских задач и онтологии программной системы. Для сравнения этих двух уровней представления знания – уровня онтологии программной системы и уровня онтологии пользовательских задач можно прибегнуть к аналогии с различием эмпирических и теоретических терминов, как оно трактуется в философии физики. Онтология пользовательских задач ассоциируется с онтологией теоретических терминов, а онтология программной системы – с онтологией эмпирических терминов.

Взаимосвязь и различие между эмпирическими и теоретическими терминами мы рассматриваем в русле работы Р. Карнапа «Философские основания физики» [2]. Эмпирическими терминами Карнап называет понятия, обозначающие феномены, которые являются либо непосредственно наблюдаемыми, либо измеряемыми сравнительно простой техникой. Это, например, «давление», «объем» и «температура», «сопротивление» и «сила тока». Смысл теоретических терминов значительно более абстрактный и гипотетический. Теоретическими терминами являются, например, «масса» и «спин электрона». Следует отметить одно исключи-

тельно важное обстоятельство, которое мы должны учитывать при рассмотрении эмпирических и теоретических терминов. Свойство «быть эмпирическим» или «быть теоретическим» здесь скорее не абсолютное, а относительное: один набор терминов является эмпирическим относительно другого; в этом случае второй набор терминов будет теоретическим относительно первого.

Сделав оговорку, что понятия эмпирических и теоретических терминов являются не абсолютными, а относительными и зависят от контекста рассмотрения, мы проанализируем в этих терминах взаимосвязь между онтологией пользовательских задач и онтологией программной системы. По сравнению с онтологией пользовательских задач описание функциональных возможностей программной системы использует эмпирические термины: действия, виды функциональности, настройки, «опции» и т.д. Описание пользовательских задач основано на теоретических терминах (опять же в сравнении с функционалом программной системы), – это цели, задачи, намерения пользователя, его потребности, предпочтения и т.п. Если мы сравним онтологию пользовательских задач и онтологию программной системы, то исходя из понятий, которые в них входят, мы можем рассматривать онтологию пользовательских задач как метауровень для онтологии программной системы. Цели, желания и задачи пользователей – это метапонятия относительно описания функциональных возможностей программной системы и способов управления ими.

Чтобы обеспечить понимание пользователем меню программы, необходима интеграция онтологии пользовательских задач и онтологии программной системы. Вопрос – каким образом эта интеграция может быть осуществлена. По существу, мы должны решить проблему «перевода» понятий онтологии пользовательских задач в понятия онтологии программной системы и обратно, в некотором смысле дать определения различным задачам пользователя на языке описания функциональности программной системы. В решении этого вопроса существует несколько программ. Одна из них связана с введением Р. Карнапом правил соответствия. Он утверждал, что нет способа, с помощью которого можно было бы определять теоретические понятия в терминах наблюдения. Но поскольку теоретические термины нужны для того, чтобы с их помощью формулировать теоретические законы, из которых, в свою очередь, получать эмпирические следствия, постольку должны быть некоторые правила перевода одних терминов в другие. Правила соответствия не являются определениями в каком-либо строгом смысле, так как не существу-

ет никаких «определений» теоретических понятий физики. Цель правил соответствия – «построить мостик» между теоретическими и эмпирическими терминами, так чтобы утверждения, записанные на одном языке, можно было транслировать на другой язык.

Сходным образом, для интеграции онтологии пользовательских задач и онтологии программной системы нам необходимы правила соответствия, устанавливающие связь между языком пользователя и языком программной системы. Но здесь возникает интересная проблема, которая позволяет по-новому взглянуть на соотношение онтологий пользовательских задач и программных систем. Если продолжать упомянутую выше аналогию, то интерес представляет проблема элиминации теоретических терминов в пользу эмпирических. Для философов науки в этом отношении важна так называемая теорема Крэйга. Согласно этой теореме, любую научную теорию T , формализованную в языке первого порядка, можно заменить на теорию T' , состоящую только из теорем первой, сформулированных исключительно в терминах языка наблюдения [3], так чтобы все наблюдаемые следствия из исходной также были следствиями последней. При этом язык исходной теории T должен эффективно делиться на два не пересекающихся, взаимно исключающих класса терминов: класс терминов наблюдения и класс теоретических терминов. Это требование явно противоречит относительности классификации терминов на теоретические и эмпирические.

Кроме того, полную редукцию теоретических терминов к эмпирическим нельзя осуществить без потери возможности индуктивной систематизации эмпирических данных. Этот результат принадлежит К. Гемпелю [4]. Продолжая нашу аналогию, отметим, что нельзя редуцировать пользовательскую онтологию к онтологии программной, потому что без первой теряется интенциональная направленность извлечения знания. Пользователь имеет целенаправленную стратегию получения определенной информации, и выстраивает процесс получения знания, с сопутствующей онтологией, в определенном порядке, который не может быть сведен к комбинаторным процессам.

Приведенная аналогия вполне оправдана. Действительно, каждый пользователь имеет свою собственную онтологию пользовательских задач и свое собственное понимание смысла понятий, принадлежащих к этой онтологии. С другой стороны, программная система предназначена для всех пользователей, – таким образом, онтология программной системы в том или ином виде должна включать в себя все возможные онтологии пользовательских задач. Это означает, что базисным уровнем

является онтология программных систем, но одновременно онтология программной системы должна соответствовать всем возможным онтологиям пользовательских задач. Другими словами, «приспособление» базисного уровня (эмпирические данные – онтология программных систем) в процессе извлечения знания к более высокому интенциональному уровню (теоретические термины – онтология пользователя) полностью аналогично предполагаемой невозможности полного устранения теоретических терминов, или же принципиальной возможности устранения пользовательской онтологии.

Кстати, именно здесь можно указать на потенциальные пути понимания сути противопоставления машины и человека как противопоставления интенциональной и комбинаторной онтологий мышления [5]. Одно из возможных решений этой проблемы – объединение пользователей в кластеры и создание различных онтологий программной системы для различных классов пользователей. Еще одна проблема, которую мы должны решить при разработке правильного пользовательского интерфейса, причем уже для данного кластера пользователей или профиля пользователя, – понятность и прозрачность дерева меню. Для того чтобы дерево меню было понятным для пользователя, необходимо как минимум выполнение следующих условий:

- пользователь должен правильно понимать все используемые ключевые понятия. Это, в частности, означает, что: 1) онтология пользователя и онтология программной системы согласованы; 2) пресуппозиции пользователя совпадают с пресуппозициями, заложенными разработчиками программной системы при создании данного варианта пользовательского меню;
- все дерево меню в целом и каждый уровень его иерархии должны быть целостными.

Наконец, третья проблема, встающая при разработке пользовательского интерфейса, – это интеграция онтологии пользовательских задач и онтологии программной системы. Как мы уже отмечали выше, интеграция пользовательской онтологии и онтологии программной системы может быть осуществлена при помощи правил соответствия.

Построение правил соответствия между онтологией пользовательских задач и онтологией программной системы может быть осуществлено следующим образом. В виде текстов естественного языка пользователи описывают имеющееся у них понимание функциональных возможностей системы. Во-первых, они излагают свое понимание иерархии функ-

циональных возможностей программы, – по существу, это описание того, каким должно быть дерево меню. Во-вторых, они описывают смысл используемых ключевых понятий, на основе которых, в частности, будет строиться интерфейс программной системы – дерево меню, инструкции, помощь пользователю и т.д. В-третьих, пользователи излагают свое понимание того, как осуществляется управление подобной программной системой, как следует задавать «опции», настройки и т.п. Из текста пользователя необходимо извлечь пресуппозиции, предварительные условия и постусловия для различных действий и процессов.

Проблемы передачи знаний: неявные утверждения и пресуппозиции

Пресуппозиции пользователей во многом определяют понимание пользователями программного интерфейса. Под пресуппозицией мы здесь понимаем утверждение X , истинность которого необходима для осмысленности и истинности утверждения Y . Рассмотрим пример – вопрос: «Иван, когда ты перестал бить свою кошку?». Этот вопрос предполагает истинность двух пресуппозиций:

У Ивана есть кошка.
Иван свою кошку раньше бил.

Вопрос имеет смысл только тогда, когда истинны обе эти пресуппозиции. Таким образом, пресуппозиции – это неявная информация, которая определяет смысл явных утверждений. При этом для гарантии ясного понимания неявные пресуппозиции желательно преобразовать в явные утверждения.

Вернемся теперь к пользовательским интерфейсам. Предельная краткость меню программной системы подразумевает наличие огромного числа пресуппозиций, знание которых необходимо для правильного понимания меню. Во-первых, пресуппозиции определяют смысл ключевых понятий, используемых при описании работы программной системы. Во-вторых, пресуппозиции соответствуют ожиданиям – тому, что пользователь ожидает найти в том или ином пункте меню. Если пресуппозиции пользователя и пресуппозиции авторов программы различны, то возникают непонимание или иллюзия понимания.

В этом месте мы наиболее отчетливо видим, что интерфейс программной системы реализует не обмен информацией, а обмен знаниями между пользователем и компьютером. Для правильного понимания

смысла данных – степени яркости, громкости и т.д. не нужны никакие presupпозиции. Для правильного восприятия и понимания знаний presupпозиции необходимы. При извлечении presupпозиций из текстов пользователей надо иметь в виду, что утверждение может быть неявным по двум причинам:

- это утверждение для пользователя является очевидным, поэтому он не видит никакого смысла указывать его явно;
- данное утверждение человек по тем или иным причинам не хочет заявлять явно.

Здесь интересно отметить связь presupпозиций и аналитических, или, более точно, концептуальных, утверждений. С одной стороны, утверждения, которые данный человек считает аналитически истинными, или истинами концептуальными, для него являются очевидными, т.е. presupпозициями. Например, никто, говоря о собаке, не станет специально уточнять, что она является животным; точно так же никто, сказав, что данный человек является холостяком, не станет добавлять, что у этого человека нет жены.

С другой стороны, presupпозиции необходимы для определения правильного смысла используемых понятий. В свою очередь, разный смысл используемых понятий задает разные множества аналитически истинных утверждений. Таким образом, различие онтологий у разных пользователей (и у разработчиков разных программных систем) напрямую связано с наличием у них различных множеств аналитически истинных утверждений и, соответственно, различных онтологий. Следовательно, один из путей реконструкции онтологии пользовательских задач, соответствующей данному кластеру пользователей, состоит в выяснении (эксplikации) presupпозиций, характерных для этих пользователей.

Для разработки методов извлечения presupпозиций мы применим подход, основанный на лингвистической теории трансформационной грамматики Н. Хомского, более конкретно, на взаимосвязи между глубинной и поверхностной структурами. Для иллюстрации этой связи мы приведем пример – последовательность предложений начиная с более поверхностных структур к более глубинным:

Кто-то взял что-то. → Мальчик взял карандаш. → Мальчик Вася взял красный карандаш. → Мальчик Вася взял красный карандаш со стола. → Мальчик Вася быстро взял красный карандаш с письменного стола.

В работе Р. Бэнндлера и Д. Гриндера проведен достаточно подробный анализ способов восстановления глубинной структуры по поверхностной [6].

Напомним, что пресуппозицией мы называем утверждение X , истинность которого необходима для осмысленности и истинности утверждения Y . Пресуппозиция – это неявная информация, которая необходима для придания смысла данному утверждению. Извлечение пресуппозиций необходимо для того, чтобы гарантировать правильное понимание, или, другими словами, для того, чтобы обеспечить одинаковое понимание у пользователя и у разработчиков программной системы. Что касается проблемы извлечения пресуппозиций из текстов пользователей для разработки адекватного интерфейса программной системы, то здесь практически всегда мы будем иметь дело с первым случаем, т.е. с теми утверждениями, которые пользователь считает совершенно очевидными. Но что очень важно: утверждения, очевидные для одних людей, для других могут быть не только неочевидными, но даже и очевидно неверными.

Для извлечения пресуппозиций, восстановления глубинной структуры по поверхностной можно применить метод ГАБЕК [7], но при этом необходимо двигаться реверсивно, в обратном направлении – от гипергештальтов к гештальтам. Наша цель – по предложению построить гештальт, который ему соответствует; по каждому элементу этого гештальта – предложению построить соответствующий ему гештальт, исходный гештальт будет гипергештальтом для данного; и так далее. Таким образом мы, двигаясь сверху вниз, строим иерархию гештальтов и гипергештальтов и одновременно строим иерархию поверхностных и глубинных структур: на каждом следующем уровне предложения будут глубинными структурами по отношению к соответствующим предложениям предыдущего уровня; все построенные предложения будут глубинными структурами по отношению к исходному предложению.

Действуя таким образом, мы для каждого кластера пользователей строим набор имеющихся у них пресуппозиций, касающихся интерфейса данной программной системы: пресуппозиций, связанных с понятиями, используемыми в меню программы, и пресуппозиций, относящихся к классу пользовательских задач, решаемых этой программой. В частности, для каждого кластера пользователей мы построим онтологию пользовательских задач – определения понятий, на языке которых пользователи описывают постановку задач и способы их решения.

Другим методом извлечения пресуппозиций является эпистемическая логика. Извлечение пресуппозиций может осуществляться самым

различным образом. В худшем случае это может быть метод проб и ошибок, в лучшем – некое теоретическое обоснование такого извлечения. Есть существенные аргументы, доказывающие, что извлечение пресуппозиций не может быть осуществлено эмпирическим образом. Одним из наиболее перспективных направлений второго рода является использование эпистемической логики. В частности, речь идет о разработанной Я. Хинтиккой и И. Халоненом теории вопросов и ответов [8]. Идея состоит в постановке вопросов и получении адекватных ответов. Под адекватностью тут понимается правильный ответ с учетом эмпирических данных. Коль скоро эмпирические данные ограничивают произвол в постановке вопросов и формулировке ответов, такое ограничение должно найти место в логической структуре процедуры вопрошания и отвечаания. Другими словами, вопрос и ответ должны включать элементы, которые определяют содержание предполагаемого знания. Элементы такого рода называются предпосылками, или пресуппозициями, вопросов и ответов.

Рассмотрим примеры процедуры получения знания путем извлечения его с помощью вопросов.

Пусть есть желание узнать, кто выиграет турнир в Уимблдоне. Итак, исследование начинается с постановки вопроса

Какой теннисист выиграет турнир?

Если речь идет о пресуппозициях как виде неявного знания, тогда ответом на этот вопрос является

(1) Я знаю, кто выиграет турнир.

Это действительно ответ на поставленный вопрос, и в этом смысле ответ является предполагаемым результатом. На самом же деле правильный ответ называется желательным результатом вопроса, и правильность его состоит в том, что он в существенной степени определяет поведение всей процедуры «вопрос – ответ». Для понимания того, как реализуется такое определение, рассмотрим «неправильный» с точки зрения эпистемической логики ответ:

(2) Я знаю, что кто-то из теннисистов Европы выиграет турнир.

«Неправильность» такого ответа состоит в том, что это не то, что хотелось знать согласно вопросу, и в этом смысле такой ответ не является желаемым. Дело в том, что ответ неоднозначен. Для получения на-

стоящего ответа требуется дополнительная информация типа. С логической точки зрения (2) не влечет (1). Очевидно, что в неоднозначном ответе предполагается следующее утверждение:

(3) Я знаю, какой именно теннисист из Европы выиграет турнир.

Это последнее и есть предпосылка ответа, который должен быть определенным или же заключительным. Хинтикка называет это условием заключительности ответа. Если мы хотим обеспечить заключительный ответ на вопрос, мы должны обеспечить истинность (3), а не только (2).

Для получения правильного ответа на поставленный вопрос, как было показано, требуется условие заключительности ответа, а именно условие (3). Но это условие не является частью эмпирических процедур познания. Дело в том, что утверждения (2) и (3) имеют разные функции. Если (2) – это способ указания на правильный объект, то (3) говорит, какой именно объект указывается. Это и есть условие заключительности ответа, т.е. условие правильного ответа. Решающий момент в подходе Хинтикки состоит в том, что (3) не является фактической истиной. Вопрос именован – вопрос семантический, или концептуальный. Если я говорю, что знаю, что существует теннисист, именуемый «Джокович», то установление имени не является вопросом эмпирическим. Это концептуальный вопрос, и, стало быть, знание, получаемое при вопросах, не является чисто эмпирическим. Более точно, получение ответа предполагает, что познающий субъект имеет определенное концептуальное знание уже перед тем, как приобрести знание эмпирическое. Еще более точно, это концептуальное знание требуется для представления эмпирического знания. Таким образом, для извлечения пресуппозиций необходим анализ, сходный с процедурами эпистемической логики.

Концептуальная часть процедуры получения знания через «вопрос-ответ» ограничена условиями заключительности ответа. Именно концептуальная часть позволяет иметь знание в том смысле, в каком понимается знание. Более точно, концептуальная часть играет роль представления знания, поскольку именно она обеспечивает заключительные ответы на исследовательские вопросы. Естественно, есть искушение отождествить концептуальную часть процедуры и ее фактическую часть с аналитической и синтетической истинами соответственно. Если так, то не является ли это искушение следствием использования весьма специфического раздела логик, а именно эпистемической логики [9]?

* * *

В настоящее время назрела необходимость перехода от онтологий к более богатым структурам. Тем не менее следует отметить, что в определенном смысле онтологии предметных областей еще не начали создаваться, как бы парадоксально это ни звучало. Тому есть две причины.

Во-первых, через некоторое время после введения онтологий в рассмотрение в контексте проблематики инженерии знаний акцент сместился с содержания онтологий предметных областей на инструментальные средства формального представления онтологий. Причем если рассмотреть практически любое из инструментальных средств представления онтологий, легко увидеть, что с помощью этих средств можно формализовать произвольную информацию о предметной области с таким же успехом, как и специфически онтологическую. Напомним, что с точки зрения специалистов по инженерии знаний, которые ввели в употребление термин «онтология предметной области», онтологическая информация – это информация о смысле ключевых терминов, на языке которых излагаются знания о предметной области.

Во-вторых, задача отделения онтологической информации о предметной области от неонтологической, т.е. задача выделения из всей информации о предметной области знаний о смысле ключевых терминов этой предметной области, является весьма нетривиальной. Эта задача не только до настоящего времени не решена, – она, по существу, еще и не поставлена. Речь идет о точной и формальной постановке задачи извлечения информации об определениях ключевых понятий, в частности об извлечении такой информации из текстов естественного языка.

Таким образом, ставятся две задачи. Более общая – это построение системы пресуппозиций, в частности извлечение пресуппозиций из текстов естественного языка. Более частная задача – разработка формальных методов представления знаний об определениях ключевых понятий (терминов) предметной области, а также разработка автоматизированных методов извлечения онтологической информации из текстов естественного языка. Под ключевыми понятиями предметной области мы здесь подразумеваем те понятия, смысл которых является специфичным для данной предметной области.

Примечания

1. См.: Пальчунов Д.Е. Моделирование мышления и формализация рефлексии I: Теоретико-модельная формализация онтологии и рефлексии // Философия науки. – 2006. – № 4

(31). – С. 86–14; *Он же*. Моделирование мышления и формализация рефлексии. Ч. 2: Онтологии и формализации понятий // Там же. – 2008. – № 2 (37). – С. 62–99; *Он же*. Решение задачи поиска информации на основе онтологий // Бизнес-информатика. – 2008. – № 1. – С. 3–13.

2. *Карнап Р.* Философские основания физики. Введение в философию науки. – М.: Прогресс, 1971.

3. *Craig W.* Replacement of auxiliary expressions // *Philosophical Review*. – 1956. – No. 65.

4. *Гемпель К.* Дилемма теоретика: исследования логики построения теории // *Гемпель К.* Логика объяснения. – М., 1998.

5. См.: *Серл Дж.П.* Сознание, мозг и программы // *Аналитическая философия: становление и развитие*. – М., 1998.

6. См.: *Бэндлер Р., Гриндер Д.* Структура магии. – СПб.: Прайм-Еврознак, 2004. – Т. 1.

7. См.: *Zelger J.* GABEK, a new method for qualitative evaluation of interviews and model construction with PC-support // *Enhancing Human Capacity to Solve Ecological and Socio-Economic Problems* / Ed. by E. Stuhler, M.O. Suilleabhain. München-Mering: Rainer Hampp Verlag, 1993. P. 128–172; *Id.* Zur Geschichte von GABEK // *GABEK II. Zur Qualitativen Forschung: On Qualitative Research* / Hrsg. R. Buber, J. Zelger. – Innsbruck; Wien-München: STUDIENVerlag, 2000. S. 13–20; *Pal'chunov D.E.* GABEK for Ontology Generation // *Learning and Development in Organizations. (GABEK – Contributions to Knowledge Organization. V. 2)* / Ed. by: P. Herdina, A. Oberprantacher, J. Zelger. – Wien: LIT, 2007. – P. 87–107.

8. *Hintikka J., Halonen I.* Semantics and pragmatics for why-questions // *Journal of Philosophy*. 1995. – V. 12, No. 92. – P. 636–657.

9. См.: *Целищев В.В.* Соотношение понятия онтологии в философии и информационных системах // *Вестник НГУ. Сер. Философия*. – 2012. – Т. 10, вып. 2. – С. 5–11.

Дата поступления 05.10.2012

Институт математики СО РАН,
г. Новосибирск
Институт философии и права
СО РАН, г. Новосибирск
director@philosophy.nsc.ru

Pal'chunov, D.Ye. and V.V. Tselishchev. The problem of knowledge retrieval in interaction between a man and a computer: ontologies and presuppositions

The paper deals with the problems of retrieval, presentation and transfer of knowledge. Two ways of knowledge exchange are considered: between people and between a man and a computer, i.e. between a software system and its user. The second way in many respects comes to the first one for the problems of incomprehension, distortion and loss of information in communication of a user with a software system are caused by different comprehension of the world, different ontologies and presuppositions which this user and designers of this software system have. Thus, the problem how to organize collaboration of a man and a computer becomes important. To solve this problem we need to solve tasks related to development of human-computer interface. Problems of human-computer interfaces are tightly connected with differences in knowledge presentation by various groups of users and software designers. In this context, study of factors determining availability or lack of common understanding of information and knowledge will be interesting.

Keywords: knowledge, information, computer, ontology, man