

М. И. ВАРСАК, В. А. ЛОВИЦКИЙ, канд. техн. наук, А. М. ЛУГАНСКИЙ

ПРЕДСТАВЛЕНИЕ И ФОРМИРОВАНИЕ ДЕКЛАРАТИВНО-ПРОЦЕДУРАЛЬНЫХ ЗНАНИЙ В ИНТЕЛЛЕКТУАЛЬНЫХ ЕСТЕСТВЕННО-ЯЗЫКОВЫХ СИСТЕМАХ

Введем понятие классов знаний интеллектуальных естественно-языковых систем (ИЕЯС) и на его базе выделим два класса декларативно-процедуральных знаний (ДПЗ), обеспечивающих практический уровень реализации ИЕЯС.

Функционирование любой ИЕЯС сводится к решению задач, формальное определение которых приведено в работах [1, 2]. Аналогично понятию «задача» определим предварительно знания ИЕЯС как четверку $DP = \langle X, Q, F, Y \rangle$, в которой X обозначает входные данные; Q — назначение и условия применимости DP ; F — конечную последовательность правил, с помощью которой на основании X получаем Y в соответствии с Q ; Y — выходные данные.

Будем считать, что ИЕЯС располагает знаниями трех типов: 1) система ЗНАЕТ ЧТО; 2) система ЗНАЕТ КАК; 3) система УМЕЕТ. Введем обозначения для описания перечисленных типов знаний. Через $Dc(W)$ обозначим декларативное описание W на языке R -выражений, где W представляет собой какой-либо факт, явление или последовательность действий; через $Pr(F)$ — программное описание какой-либо последовательности действий на одном из алгоритмических языков, оформленное в виде программного модуля. Тогда в соответствии с введенными обозначениями всевозможные комбинации трех типов знаний можно наглядно представить в виде таблицы, в которой 1 и 0 обозначают наличие или отсутствие соответствующего типа знаний в базе ИЕЯС. Полный перебор позволяет выделить семь классов знаний.

Классы знаний	Система ЗНАЕТ ЧТО	Система ЗНАЕТ КАК	Система УМЕЕТ
	$Dc(X, Q, Y)$	$Dc(F)$	$Pr(F)$
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Класс 1 составляет активные процедуральные знания (АПЗ), поскольку конечная последовательность правил F в $Pr(F)$ представлена на таком языке, который позволяет ИЕЯС выполнить F . Оформление F в виде программного модуля предполагает его реализацию на уровне подпрограммы (процедуры) или на уровне фазы.

Будем различать непосредственно выполнимые АПЗ (НВАПЗ) и потенциально выполнимые АПЗ (ПВАПЗ). К НВАПЗ относятся только такие $Pr(F)$, которые представлены корневыми фазами, а к ПВАПЗ — подпрограммы (процедуры) или фазы. Для выполнения НВАПЗ достаточно их загрузить в оперативную память ЭВМ и дать команду на выполнение. Для выполнения же ПВАПЗ их необходимо или преобразовать в (дополнить до) НВПЗ, или найти такие НВАПЗ, которые вызывают эти ПВАПЗ.

НВАПЗ или ПВАПЗ, которые вызывают другие ПВАПЗ, называются процедуральными метазнаниями (ПМЗ). ПМЗ показывают, какие процедуральные знания и в какой последовательности нужно использовать для получения Y на основании X в соответствии с Q .

Например, одно и то же F можно представить в виде различных вариантов описаний на Фортране:

$Pr_1(F)$	$Pr_2(F)$	$Pr_3(F)$
$A = 5.7$	<i>SUBROUTINE PRM (A, B)</i>	$A = 5.7$
$B = -1.6$	$P = A$	$B = -1.6$
$P = A$	$A = B$	<i>CALL PRM (A, B)</i>
$A = B$	$B = P$	<i>STOP</i>
$B = P$	<i>RETURN</i>	<i>END</i>
<i>STOP</i>	<i>END</i>	
<i>END</i>		

$Pr_1(F)$ представляет собой НВАПЗ; $Pr_2(F)$ — ПВАПЗ $Pr_3(F)$ — это ПМЗ в виде НВАПЗ. Более того, одни и те же ПЗ на одном и том же языке могут быть описаны по-разному. Например,

```

 $Pr_4(F)$ 
SUBROUTINE PRM1 (A, B)
 $A = A + B$ 
 $B = A - B$ 
 $A = A - B$ 
RETURN
END

```

Итак, завершая описание 1-го класса значений ИЕЯС, отметим, что если система располагает только данным классом знаний, то она самостоятельно их активизировать не сможет.

В самом деле, даже если представить, что ИЕЯС, последовательно перебирая оглавление библиотеки загрузочных модулей, извлекает корневые фазы и выполняет (активизирует) их, то и в этом случае система не только должна располагать соответствующим ПМЗ, с помощью которого будет осуществляться поиск, извлечение и выполнение соответствующих НВАПЗ, но и уметь к нему обращаться, т. е. располагать соответствующим $Dc(X, Q, Y)$.

Класс 2 включает в себя пассивные (по отношению к ИЕЯС, которая располагает этими знаниями) процедуральные знания, т. е. конечная последовательность правил F записана на таком языке, который не позволяет данной ИЕЯС_{*i*} выполнить F . Но в этом случае обязательно должна существовать такая внешняя ИЕЯС_{*j*}, которая может выполнить F . Иными словами, $Dc(F)$ для ИЕЯС_{*i*} должно быть $Pr(F)$ для ИЕЯС_{*j*}.

Привлечение естественного языка для описания F приводит к случаям многозначного описания одного и того же F . Например, для рассматриваемого F имеем:

- | | |
|--|---|
| $Dc_1(F)$ | $Dc_2(F)$ |
| 1. $\langle A \rangle$ занести в $\langle P \rangle$. | 1. $\langle P \rangle$ присвоить значение $\langle A \rangle$. |
| 2. $\langle B \rangle$ занести в $\langle A \rangle$. | 2. В $\langle A \rangle$ записать $\langle B \rangle$. |
| 3. $\langle P \rangle$ занести в $\langle B \rangle$. | 3. В $\langle B \rangle$ занести $\langle P \rangle$. |

Если система располагает только данным классом знаний, то она не имеет доступа ни к $Dc(F)$, ни к $Dc(f_i)$, а следовательно, не может определить идентичность $Dc_1(F)$ и $Dc_2(F)$. Поскольку в работе для декларативного описания F привлекается только естественный язык, то вариант многозначного декларативного описания F с помощью различных алгоритмических языков здесь не рассматривается.

Класс 3 объединяет ПЗ предыдущих двух классов и представляет собой полные процедуральные знания. Принципиально важным является рассмотрение вопроса соотношения между $Dc(F)$ и $Pr(F)$. Охарактеризуем F : 1) F — это абстрактное понятие, материальное представление которого существует только в виде $Dc(F)$ и $Pr(F)$; 2) поскольку F — это конечная последовательность правил, то обозначив каждое правило через f_i , получим: $Dc(F) = Dc(f_1) \rightarrow Dc(f_2) \rightarrow \dots \rightarrow Dc(f_i) \rightarrow \dots \rightarrow Dc(f_n)$ или $Pr(F) = Pr(f_1) \rightarrow Pr(f_2) \rightarrow \dots \rightarrow Pr(f_i) \rightarrow \dots \rightarrow Pr(f_n)$, где символ \rightarrow обозначает операцию следования; 3) с помощью последующих определений введем различные отношения между $Dc(F)$ и $Pr(F)$.

Определение 1. Будем говорить, что между описаниями $Dc(F)$ и $Pr(F)$ существует взаимно-однозначное соответствие, если

$$1) Dc(F^Q)(X) = Pr(F^Q)(X);$$

$$2) а) \forall Dc(f_i) \in Dc(F) \exists Pr(f_i) \in Pr(F) \exists T_s(T_s(Dc(f_i)) = (Pr(f_i))) \vdash T_s(Dc(F)) = (Pr(F));$$

$$б) \forall Pr(f_i) \in Pr(F) \exists Dc(f_i) \in Dc(F) \exists T_a(T_a(Pr(f_i)) = (Dc(f_i))) \vdash T_a(Pr(F)) = (Dc(F)),$$

где символ « \in » обозначает вхождение элемента $Dc(f_i)$ в последовательность $Dc(F)$.

Первый пункт данного определения говорит о том, что некоторая внешняя по отношению к ИЕЯС система, выполнив $Dc(F)$ для X в соответствии с Q , получит такой же результат, что и ИЕЯС, выполнив $Pr(F)$ для тех же X и Q . Пункт 2, а указывает на возможность ИЕЯС с помощью транслятора T_s преобразовывать каждое $Dc(f_i)$ в $Pr(f_i)$, а следовательно, и $Dc(F)$ в $Pr(F)$, т. е. наличие транслятора T_s позволит ИЕЯС по декларативному описанию F синтезировать активные ПЗ. Пункт 2, б говорит о возможности ИЕЯС с помощью ретранслятора T_a преобразовывать активные ПЗ в соответствующие декларативные описания. Выполнение только одного из пунктов

(2, а или 2, б) говорит об однозначном соответствии между описаниями $Dc(F)$ и $Pr(F)$. Взаимно однозначное соответствие между $Dc(F)$ и $Pr(F)$ будем обозначать через $Dc(F) \leftrightarrow Pr(F)$, а однозначные соответствия — через $Dc(F) \rightarrow Pr(F)$.

Определение 2. Будем называть $Dc(F)$ декларативными процедуральными метазнаниями, если $Dc(F) \rightarrow Pr(F)$.

Определение 3. Будем называть $Pr(F)$ процедуральными знаниями, «осознанными» ИЕЯС, если $Dc(F) \leftrightarrow Pr(F)$.

Определение 4. Будем говорить, что описания $Dc(F)$ и $Pr(F)$ независимы, если:

$$1) Dc(F^Q)(X) = Pr(F^Q)(X);$$

$$2) а) \neg \forall Dc(f_i) \in Dc(F) \exists Pr(f_j) \in Pr(F) \exists T_s(T_s(Dc(f_i)) = (Pr(f_j))) \nabla \forall Dc(f_i) \in Dc(F) \forall Pr(f_j) \in Pr(F) \neg \exists T_s(T_s(Dc(f_i)) = (Pr(f_j)));$$

$$б) \neg \forall Pr(f_i) \in Pr(F) \exists Dc(f_j) \in Dc(F) \exists T_o(T_o(Pr(f_i)) = (Dc(f_j))) \nabla \forall Pr(f_i) \in Pr(F) \forall Dc(f_j) \in Dc(F) \neg \exists T_o(T_o(Pr(f_i)) = (Dc(f_j))),$$

где символ « ∇ » обозначает знак операции «разделительно ИЛИ». Независимость описаний $Dc(F)$ и $Pr(F)$ будем обозначать как $Dc(F) \perp Pr(F)$.

Например, зададим F_1 и F_2 в виде трех описаний $Dc(F_1)$, $Pr(F_1)$ и $Pr(F_2)$:

	$Dc(F_1)$	$Pr(F_1)$	$Pr(F_2)$
1.	$\langle A \rangle$ занести в $\langle P \rangle$.	$P = A$	$A = A + B$
2.	$\langle B \rangle$ занести в $\langle A \rangle$.	$A = B$	$B = A - B$
3.	$\langle P \rangle$ занести в $\langle B \rangle$.	$B = P$	$A = A - B$

Очевидно, $Dc(F_1) \rightarrow Pr(F_1)$ и $Dc(F_1) \rightarrow Pr(F_2)$, несмотря на то, что назначение и F_1 , и F_2 — одинаковое, а именно: «содержимое ячеек $\langle A \rangle$ и $\langle B \rangle$ поменять местами».

Итак, рассмотренные классы знаний из-за отсутствия декларативного описания не могут быть активизированы системой самостоятельно, поэтому в плане обсуждаемой проблемы представляют собой чисто теоретический интерес.

Класс 4. К этому классу знаний относятся декларативные описания конечных множеств X , Q и Y . Иными словами, утверждается, что любые декларативные знания ИЕЯС представляют собой описания X , Q и (или) Y соответствующих процедуральных знаний. Декларативные знания вводятся в систему на естественном языке в виде отдельных предложений или связанных текстов. Для каждого предложения в активном режиме системой формируется семантико-прагматическая окрестность ситуации (СПОС), которая включает в себя следующие семантические отношения (СМНО): 1) «цель» системы; 2) «цель» субъекта действия; 3) «цель» источника информации (пользователя); 4) «синоним», «элемент класса», «часть целого».

и «противоположно» по отношению к отдельным словоформам и к целым предложениям; 5) «имя» субъекта или «название» объекта; 6) «назначение» объекта; 7) «условие» выполнения действия; 8) «причина» возникновения действия, его выполнения или завершения; 9) «результат» действия; 10) «оценка» действия; 11) «характеристика» субъектов, объектов и действий. СМНО «валентность» (модель управления) для каждой словоформы формируется системой самостоятельно. Итак, с помощью отдельных простых распространенных предложений задаются программные модули, имена которых совпадают с глаголами этих предложений, а сформированные СПОСы определяют их назначение, условие применимости и результат их действия. Например, предложение «Робот взял деталь с полки» вместе с соответствующим этому предложению СПОСом определяет естественноречевой программный модуль, имя которого «взять».

При описании искусственных программных модулей, $Pr(F)$ которых задается на каком-либо алгоритмическом языке, добавляются такие СМНО, как «язык», «спецификация» («вид», «тип», «длина»), «значение», «измерение».

Важно подчеркнуть, что и этот класс знаний имеет чисто теоретическое значение, поскольку даже ввод X невозможен без существования соответствующего $Pr(F)$.

Класс 5 представляет собой активные декларативно-процедуральные знания (АДПЗ), т. е. $DP_i = \langle Dc(X_i, Q_i, Y_i), Pr(\bar{F}_i) \rangle$.

Будем считать, что на вход ИЕЯС могут подаваться только естественно-языковые высказывания (ЕЯВ), обозначаемые через S . Особенность данного класса знаний состоит в том, что одно и то же ЕЯВ и его компоненты выступают в роли различных составляющих декларативной части описания DP . В зависимости от этого будем различать различные подклассы АДПЗ. Первый подкласс составляют «врожденные» АДПЗ (DP^B). Их соотношенность с входными ЕЯВ характеризуется следующим выражением:

$$\forall S_i \exists DP_i^B ((S_i = X_i) \& (Y_i = Dc(X_\mu, Q_\mu, Y_\mu))).$$

Например, с помощью DP_i^B для $S_i = \langle \text{Робот взял деталь с полки} \rangle$ формируется $Dc(X_\mu, Q_\mu, Y_\mu)$ в виде СПОСа, где словоформа «взять» представляет собой имя соответствующего $Pr(F_\mu)$, которое, возможно, и не известно ИЕЯС.

Активизировать «врожденные» АДПЗ могут только начальные (или пусковые) АДПЗ (НАДПЗ), составляющие отдельный подкласс. Никакое декларативное описание НАДПЗ в ИЕЯС не вводится, кроме их имен, с помощью которых внешняя (по отношению к ИЕЯС) система, зная $Dc(X, Q, Y)$ НАДПЗ, на декларативном уровне активизирует соответствующие НАДПЗ. Например, для того чтобы активизировать НАДПЗ ЭВМ типа ЕС, оператор устанавливает пакет дисков с резиденцией

системы на выбранное устройство, набирает адрес этого устройства на пульте ЭВМ (задает X) и по имени (нажимает кнопку ЗАГРУЗКА) вводит в память соответствующую НАДПЗ.

Третий подкласс АДПЗ составляют сформированные (или приобретенные) АДПЗ, начальная активизация которых осуществляется врожденными АДПЗ, т. е.

$$\exists Dc(X_{\mu}, Q_{\mu}, Y_{\mu}) \exists DP^B((X_{\nu} = Qc(X_{\mu}, Q_{\mu}, Y_{\mu})) \& (Y_{\nu} = Pr(F_{\mu})) \vdash Pr(F_{\mu}^{Q_{\mu}})(X_{\mu}) = Y_{\mu}.$$

Квантор существования перед $Dc(X_{\mu}, Q_{\mu}, Y_{\mu})$ говорит о том что декларативно-процедуральные знания системы могут содержать только декларативные компоненты не существующих DP_i , но которые обязательно должны представлять собой X для существующих в ИЕЯС DP_j .

Класс 6 включает в себя пассивные декларативно-процедуральные знания, т. е. $DP_i = \langle Dc(X_i, Q_i, Y_i), Dc(F_i) \rangle$.

Существование ретранслятора T_B для $Pr(F)$ или транслятора T_s для $Dc(F_i)$ позволяет получить полные ДПЗ, составляющие седьмой класс знаний интеллектуальной естественно-языковой системы.

Таким образом, только пятый и седьмой классы знания обеспечивают практический уровень, реализации на ЭВМ интеллектуальной диалоговой естественно-языковой системы.

Отличительной чертой ИЕЯС является их способность расширять свои функциональные возможности по декларативному описанию новых функций активных процедуральных знаний. Это возможно только в том случае, когда во-первых, система располагает полными ДПЗ, а во-вторых, все те активные процедуральные знания, композиция которых приведет к расширению функциональных возможностей системы, связаны с соответствующими пассивными процедуральными знаниями взаимно-однозначным соответствием.

Практическая ценность ИЕЯС в значительной степени будет определяться тем, какими декларативными метазнаниями будет располагать система. Будем различать три типа декларативных метазнаний системы: 1) «система знает, что кто-то ЗНАЕТ ЧТО (или где-то можно УЗНАТЬ ЧТО)»; 2) «система знает, что кто-то ЗНАЕТ КАК (или где-то можно УЗНАТЬ КАК)»; 3) «система знает, что кто-то УМЕЕТ (или где-то УМЕЮТ)».

Перечисленные и рассмотренные типы и классы знаний (с учетом сделанных выводов) составляют основу базы знаний системы ДЕСТА-3, представляющей собой качественное функциональное расширение системы ДЕСТА-2 [3].

Список литературы: 1. Версак М. И., Ловицкий В. А. Вопросы анализа и синтеза задачно-решающей системы оперативного управления основным производством//АСУ и приборы автоматки.—1975.—Вып. 34.—С. 114—121. 2 Бондарев В. М., Ловицкий В. А. Классификация интеллектуальных задач//Пробл.