А. Р. МАРТИРОСЯН

# ПРОГРАММА, АНАЛИЗИРУЮЩАЯ ЗАПРОСЫ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

## Введение

Расширение сферы применения ЭВМ делает настоятельным создание для пользователей естественных условий общения с ЭВМ. Важное значение имеет предоставление пользователям возможности общения с ЭВМ на естественном языке (ЕЯ). Системы, предоставляющие такую возможность, должны опираться на модель языка и модель представления внешнего мира, так как только при наличии общности языка и знаний об окружающем мире можно говорить об общении.

Усилия большинства лингвистов, занимавшихся данной проблемой, были сосредоточены в основном на «универсальном» языке. Такие работы представляют безусловный интерес и являются необходимым этапом к построению всеобъемлющей системы, понимающей естественный язык (СПЕЯ). В настоящее время эти работы носят экспериментальный характер и отличаются крайне ограниченной предметной областью (ПО) [1—3]. Для СПЕЯ, работающих в реальных и неупрощенных ПО, соответствующая модель языка становится достаточно сложной как в теоретическом, так и в техническом аспекте.

Предлагается описание модели языка, предназначенного для систем, обеспечивающих пользователям доступ к базам данных (БД) на ЕЯ. Отличительной особенностью предлагаемой модели является то, что знания о ПО естественным образом представляются в модели языка. Эта модель реализована в разработанном в филиале ВНЦХ Анализаторе запросов на ЕЯ к БД по историям болезней. В дальнейшем в качестве примеров будут использоваться запросы к этой БД.

## Модель языка

Описываемая система предоставляет неподготовленным пользователям возможность формулировать запросы к БД на ЕЯ без каких-либо грамматических ограничений на структуру запроса. Мы предполагаем, что диалог на ЕЯ является не самоцелью, а одним из возможных способов доступа пользователей к данным. Возможно также получение данных и традиционным способом, с помощью специально разработанных прикладных программ, в частности, диалога типа «меню». Основной целью нашей работы является не построение СПЕЯ, а предоставление пользователям эффективных средств доступа к хранимым данным. Из-за избыточности и расплывчатости ЕЯ возможны ситуации, когда общение на ЕЯ будет не самой лучшей формой доступа к данным. Реакцию системы в этом случае намного труднее проследить и понять, чем в случае выполнения формализованных запросов. Это может привести к недоразумениям и неоправданным надеждам. Поэтому, учитывая требования эффективности и удобства для пользова-

телей, часть возможных запросов к БД реализована в диалоге типа «меню». Это позволило сделать Анализатор запросов на ЕЯ более простым, гибким и эффективным.

При разработке Анализатора мы придерживались того мнения, что процесс общения между людьми является не столько процессом передачи смысла от одного участника к другому, сколько процессом, в котором участники преследуют свои цели [4]. Фактически цели, преследуемые участниками общения, и определяют структуру диалога. В нашем случае целью пользователей является получение каких-то данных, которые явно или неявно хранятся в БД. Пользователю также известно, что единственной целью системы является предоставление данных. Любой осмысленный запрос пользователя должен своей формой и содержанием учитывать эти цели. И именно на анализ запросов такой структуры ориентирован Анализатор.

С нашей точки зрения модель языка, принятая в системе, обеспечивающей доступ к БД на ЕЯ, должна удовлетворять следующим требованиям:

— накладываемым на ЕЯ ограничением являются в первую очередь ограничения ПО, отраженной в БД, и ее лексического состава;

— пользователи в основном не имеют навыков работы с печатающей клавиатурой и естественно, что в запросах будут встречаться орфографические ошибки. В первую очередь это будут чисто механические ошибки типа опечаток, пропуска букв или слов, ошибки несогласования и т. д. Анализатор должен разумно реагировать в каждом таком случае, пытаться в диалоге скорректировать правописание и восстановить пропущенные знаки;

— система должна быть в состоянии понимать грамматически некорректно составленные запросы;

— способность адаптации к профессиональному жаргону данной ПО. Это связано с тем, что из-за избыточности ЕЯ пользователи при работе с терминалом будут стремиться к сокращениям и предпочтут обходиться, где это возможно, без полных предложений.

При построении модели языка, в рамках которого интерпретируются запросы к БД, мы исходили из следующих предположений:

— каждая БД характеризуется определенным множеством семантических составляющих. Пользователи при формулировании запросов оперируют понятиями из этого множества. Для БД по историям болезни примерами семантических составляющих являются НОМЕР ИСТОРИИ БОЛЕЗНИ, ДАТА ПОСТУПЛЕНИЯ (больного), ОТДЕЛЕНИЕ, ДАТА ОПЕРАЦИИ, ЛЕЧАЩИЙ ВРАЧ и т. д.

— если распознаны все семантические составляющие, использованные в запросе, то этого достаточно для однозначной интерпретации запроса. Например, запросы: «Кого из отделения нефрологии сегодня оперировали» и «Оперировали из отделения нефрологии кого сегодня» будут проинтерпретированы однозначно, хотя последний запрос грамматически некорректен.

— любой осмысленный запрос к БД—это требование какой-то информации, выводимой из атрибутов БД.

Знания о языке можно разделить на общее знание, представленное в виде грамматики, и индивидуальное знание, представленное в виде словарной информации. Анализатор запросов состоит из управляющей программы и словарей. В управляющей программе отражены знания системы о грамматике ЕЯ с учетом специфики предложений, выражающих запрос к БД. Отметим, что любое анализируемое системой предложение является не каким-то абстрактным предложением, а имеет определенную ситуативную привязку, так как является запросом к

конкретной БД. Мы придерживаемся также того взгляда, что смысл предложения—это выражение соотнесенной с ним ситуации, а смысл слов, входящих в данное предложение, есть результат разложения смысла предложения [4]. Под толкованием слова, которое приписывается слову в языке, понимается тот «псевдосмысл», который образуется на основании наиболее частого использования слова в предложениях. Таким образом, смысл слова, но не толкование, присваивается слову динамически, в процессе интерпретации предложения, а не статически, то есть заранее не известен. А так как в каждом конкретном случае известна ситуативная привязка, то целесообразнее хранить в словаре смысл слова, а не его толкование, как это обычно делается в СПЕЯ. В этом случае можно даже утверждать, что модель ПО с учетом целей пользователей будет отражена в словаре.

Такая структура словаря значительно облегчает процесс интерпретации запросов, хотя и делает словарь достаточно сложным. С другой стороны, упрощается управляющая программа, что в целом позволяет системе сравнительно безболезненно адаптироваться как к новой ПО, так и к языку пользователя.

## Анализ запроса

Обычно в СПЕЯ есть специальная фаза морфологического анализа, проводимая с более или менее достаточной глубиной. Задачей этой фазы является обработка словоформ вне контекста, то есть, во-первых, идентификация словоформы, а во-вторых, приписывание ей характеризующего ее комплекса морфологической информации. Это возможно сделать двумя способами: процедурным или декларативным. При процедурном способе используется словарь основ. Анализ состоит в выделении основы из словоформы, ее идентификация в словаре основ и приписывание словоформе соответствующей морфологической информации. При декларативном способе используется словарь словоформ, в котором хранятся все возможные словоформы каждого слова с приписанной им морфологической информацией. Анализ фактически состоит в поиске словоформы в словаре и выборе соответствующей информации. Вследствие этого декларативный способ работает быстрее процедурного. К недостаткам декларативного относится необходимость хранения всех словоформ и соответствующей им морфологической информации, что приводит к большой трудоемкости при заполнении такого словаря.

Нами используется модифицированный вариант декларативного подхода. Его суть заключается в следующем. При анализе запроса мы исходим из того, что чисто морфологическая информация не является столь существенной для понимания смысла запроса. Поэтому в явном виде морфологическую информацию можно и не хранить в словаре. В этом случае целесообразно иметь в системе два словаря: основ и словоформ. В словаре основ содержится смысл слов, а в словаре словоформ имеется только ссылка на соответствующий элемент в словаре основ. Нет особой необходимости предварительного ввода в словарь всех словоформ слова. Это легко будет делаться автоматически в процессе работы системы. Кроме того, словарь не будет «засорен» словоформами, которые никогда не встречались в запросах. Как показано в [5], лексический состав для каждой ПО содержит около 1000—1500 слов. Среди этих слов можно было бы выделить слова, не зависящие от конкретной ПО, например, вопросительные, и предварительно описать их в словаре основ. Однако их количество, на наш

взгляд, небольшое, что и делает это излишним. Таким образом, в предлагаемом подходе, даже при нынешнем уровне развития технических средств, размеры словарей являются вполне приемлемыми.

Одной из центральных компонент системы является представление смысла слов. При описании этого представления мы будем придерживаться текущего состояния системы. В системе различается 20 классов слов. При разбиении на классы мы придерживались скорее прагматических соображений, чем теоретических. Ниже приводится описание некоторых классов.

Слова-условия: Для каждой конкретной БД некоторые слова являются выражением логического условия. Например, слово «женщина» является условием ПОЛ БОЛЬНОГО=Ж. Здесь ПОЛ БОЛЬНОГО— это имя атрибута БД. Возможно также объединение с помощью конъюнкции и/или дизъюнкции нескольких условий.

Слова-атрибуты: Они являются наименованием атрибутов БД. Для них указывается семантический признак (ЧЕЛОВЕК, ДАТА, МЕСТО и т. д.) и допустимые операции сравнения. Например, для атрибута ПОЛ БОЛЬНОГО бессмыслена операция больше или меньше.

Слова-значения: Они обычно являются значениями атрибутов БД. Другим способом применения является ссылка, что оно является началом некоторого словосочетания. После того как специальная программа осуществит «сборку» словосочетания, словосочетание может сослаться на некоторый элемент словаря основ, принадлежащий любому из определенных в системе классов.

Слова-действия: Их можно, с определенной натяжкой, описать как семантическую компоненту модели управления этого слова. С каждым таким словом связываются тройки вида вопросительное слово—атрибут/ы/ БД—семантический признак атрибута. Например, для слова «поступил» это будут КТО—ФИО БОЛЬНОГО—ЧЕЛОВЕК, КУДА—ОТДЕЛЕНИЕ—МЕСТО, КОГДА—ДАТА ПОСТУПЛЕНИЯ—ДАТА и т. д.

Мы считаем, что система поняла запрос, если она, получив на вход запрос на ЕЯ, преобразовала его в выражение F(Z)/W, где Z—список атрибутов БД, необходимых для вывода ответа, F—функция вывода ответа, W—логическое выражение, состоящее из троек вида A—C—B и A—C—V, где A и B—атрибуты БД, C—условие (равно, больше и т. д.), V—константа.

Система ориентирована на обработку изолированных запросов. Запрос анализируется пословно. Каждое слово ищется в словаре словоформ. Если поиск кончается неудачей, система вступает в диалог с пользователем, пытаясь выяснить его. Сначала пытается выяснить, является ли оно значением какого-либо атрибута БД. В этом случае система может, по желанию пользователя, ввести это слово в словарь основ. Отметим, что оно вводится также и в словарь словоформ. Во втором случае система пытается узнать некую каноническую форму этого слова. Затем слово заносится в словарь словоформ. Фактически это является обучением новым словоформам и синонимичным словам. В третьем случае оно может игнорироваться и в дальнейшем рассматриваться как «шум», или вообще может быть прекращен дальнейший анализ запроса. Заметим, что во всех этих случаях новые слова заносятся также и в специальный файл, который периодически просматривается администратором системы. В случае необходимости у него имеются средства для корректировки словарей.

После распознания всех словоформ происходит «сборка» словосочетаний. Это достигается сравнительно легко вследствие наличия прямых ссылок в словаре основ. Каждое словосочетание, как и все сло-

ва, не вошедшие в словосочетание, замещаются соответствующей словарной статьей из словаря основ.

Управляющая программа состоит из набора контекстных регистров (14 регистров), реализованных в виде стеков, и программ, обрабатывающих каждый класс слов. При анализе очередного слова вызывается соответствующая программа, которая на основе словарной информации и текущих значений контекстных регистров вносит изменения в регистры. Основными принципами при анализе запроса являются принцип «активизации» и «ожидания». Определенные слова могут «активизировать» связанные с ним семантические составляющие. Некоторые слова указывают на определенные характеристики последующих слов. Например, в запросе:

«Кто поступил в январе в отделение нефрологии?» «поступил» активизирует семантическую составляющую «ДАТА ПОСТУПЛЕНИЯ». При анализе словоформы «январе», которая имеет семантический признак «ДАТА», она будет трактоваться как «ДАТА ПОСТУПЛЕНИЯ». За предлогом «в» система ожидает, что будет следовать слово, имеющее семантический признак «ДАТА» или «МЕСТО».

После того как проанализированы все слова, система на основании заполненных контекстных регистров может определить бессмысленные запросы. Для осмысленных запросов строится соответствующее выражение. Система реализована на версии 4.1 ОС ЕС ЭВМ, язык программирования—PL/1, количество операторов—1400.

Ա. Ռ. ՄԱՐՏԻՐՈՍՅԱՆ

## ԲՆԱԿԱՆ ԼԵԶՎՈՎ ՊԱՀԱՆՋԱՐԿՆԵՐԸ ՎԵՐԼՈՒԾՈՂ ԾՐԱԳԻՐ

Ամփոփում

*Առաջարկվում է բնական լեզվի մոդելի նկարագրություն, որը կարող է կիրառվել բնական լեզվի օգնությամբ տվյալների բազայից ինֆորմացիայի ստացման համար նախատեսված սիստեմներում։ Մոդելի առանձնահատկությունը կայանում է նրանում, որ առարկայական տիրույթի մասին գիտելիքները ներդրնակորեն ընդգրկվում են լեզվական մոդելում։ Ներկայացվում է այս մոդելը իրացնող սիստեմ։*

## ЛИТЕРАТУРА

1. *Т. Виноград.* Программа, понимающая естественный язык. М., Мир, 1976.
2. *Р. Шенк.* Обработка концептуальной информации. М., Энергия, 1980.
3. *Дж. Майлопулос* и др. TORUS—Система для управления данными, понимающая естественный язык.—В кн.: Труды IV Международной объединенной конференции по искусственному интеллекту, т. 6, 1975.
4. *Э. В. Попов.* Общение с ЭВМ на естественном языке. М., Наука, 1982.
5. *А. Малхотра.* Использование в управлении естественно-языковых систем, обладающих знанием. Анализ требований.—В кн.: Труды IV Международной объединенной конференции по искусственному интеллекту, т. 6, 1975.