

# Интеллектуальные модули для медицинских приложений

Ю. Прокопчук, А. Алпатов, О. Харченко,  
Н. Огданский, В. Костра

## Введение

Многолетний опыт разработки и эксплуатации медицинских информационных систем (МИС) [1]–[13] позволил выделить ряд интеллектуальных модулей, от характеристик которых во многом зависит эффективность работы МИС в целом. К числу таких модулей можно отнести:

- *Конструктор шаблонов / документов*, описывающих различные объекты и процессы предметной области (шаблоны являются прототипами документов). Конструктор выполняет три основные функции: создание шаблона; создание документа на основе шаблона; визуализация документа, хранящегося в архиве. Шаблоны документов входят в состав метаинформации [9].
- *Лексический процессор*, использующий заготовки словоформ ограниченного профессионального языка для создания текстовых описаний характеристик объектов и процессов и формирования запросов к базе данных. Лексический процессор позволяет также выявлять факторы риска развития угрожающих состояний. Описание стандартной лексики входит в состав метаинформации [3].
- *Семантический процессор*, базовые конструкции которого позволяют формировать заключения на основе семантических правил (операций) отображения реальных значений факторов,

фигурирующих в документе, на множество числовых и лингвистических переменных. Семантический процессор позволяет также вводить в документы вычисляемые поля и выделять факторы риска развития угрожающих состояний. Описание семантических операций входит в состав метаинформации [12].

- *Проблемный решатель*, использующий информацию, полученную, в частности, от лексического и семантического процессоров (факторы риска), для решения задач диагностики, прогнозирования и выбора такого уровня помощи, при котором минимизируется вероятность реализации угрозы. Причинно-следственные схемы явлений, правила образования производных понятий (синдромов, заключений) входят в состав метаинформации, образуя базу знаний МИС [6, 7, 9].

Объекты, с которыми оперируют конструктор шаблонов и лексический процессор, представляют собой синтаксические знания интеллектуальной управляющей системы, а объекты и события, с которыми оперируют семантический процессор и проблемный решатель – семантические и прагматические знания.

В основе построения МИС лежит понятие «электронная медицинская карта» (ЭМК), рассматриваемая как кортеж упорядоченных во времени первичных электронных документов  $D_i$  и вторичных документов  $V_j$  (обзоров): ЭМК =  $\langle D_1, D_2, \dots, D_n; V_1, V_2, \dots, V_m \rangle$ . К первичным документам относятся результаты исследований и консультаций, дневники, хирургическая документация, первичный осмотр, план лечения и обследования и т.д. К вторичным – сводные данные, температурный лист (данные берутся из дневников), статкарта, просмотр данных в динамике, экспертные (прогнозные) оценки и т.д. Документы могут быть нескольких типов, например: исследования, хирургическая документация, эпикризы и т.д. Для каждого типа документа имеется свой иерархический справочник, в котором описаны ряд характеристик документов и содержатся ссылки на таблицы базы данных, описывающих шаблоны документов. С каждым полем документа могут быть связаны справочник, лексическое дерево, представляющее собой фрагмент ограниченного профессионального языка, или блок семантических операций.

В основе любого первичного медицинского документа лежит некоторый смысловой объем данных о пациенте, который обозначим через  $D_p$ . Каждому  $D_p$  можно поставить в соответствие множество входных форм  $\{F_{inp}\}$ , множество выходных форм  $\{F_{out}\}$ , множество представлений на уровне баз данных  $\{P_{DB}\}$ . При этом часть данных вводится пользователем вручную ( $D_u$ ), часть рассчитывается с помощью жестко «зашитых» в рабочую программу процедур ( $D_a$ ), а часть определяется с помощью семантических операций ( $D_s$ ), являющихся частью метаданных. Таким образом, можно записать:  $D_p = \langle D_u, D_a, D_s \rangle$ . Ниже описана технология стандартизации и ускорения ввода блока данных  $D_u$  на основе использования ограниченного профессионального языка и лексического процессора, а также механизм реализации семантических операций, формирующих блок данных  $D_s$ .

## 1. Описание Ограниченного Профессионального Языка

Под Ограниченным Профессиональным Языком (ОПЯ) будем понимать множество лексических деревьев, описывающих объекты, процессы, события данной предметной области. Лексическое дерево (ЛД) может иметь несколько представлений: текстовое, машинное, визуальное. Текстовое представление хранится в обычном текстовом файле (в кодировке DOS). Оно не является обязательным, но весьма удобно при первом построении ЛД или его существенной модернизации. В дальнейшем текстовое представление ЛД должно быть преобразовано в машинное представление – специальным образом организованную базу данных. В процессе работы пользователь непосредственно взаимодействует с визуальным представлением ЛД, которое определяется интерфейсом лексического процессора, осуществляющего навигацию по лексическому дереву и выбор необходимых лексем. Лексический процессор загружает машинное представление ЛД, позволяет его корректировать (минуя текстовое представление) и выполнять конвертацию текстового представления в машинное и обратно (подобное преобразование не является взаимнооднозначным).

При формализации ОПЯ разработчики стремились к максимальной выразительности синтаксических конструкций при минимальном использовании вспомогательных символов. Главное требование к ОПЯ можно сформулировать следующим образом: итоговый текст, сформированный в результате использования ОПЯ, должен в точности соответствовать принятому «ручному» стилю описания данного объекта или процесса. Необходимо отметить, что «ручной» стиль глубоко индивидуальный у каждого специалиста. В итоге, под лексическим деревом (текстовое представление – индекс  $t$ ) будем понимать упорядоченное множество двухэлементных кортежей, удовлетворяющих определенным условиям:

$ЛД_t = \{W_i = \langle a_i, L_i \rangle \mid a_i \in N, L_i - \text{словоформа}, i = 1..n; S(W_1, \dots, W_n) - \text{множество синтаксических и семантических ограничений}\}$ .

Словоформа (лексема)  $L$  представляет собой кортеж знаков, включающий пробелы, а также специальные управляющие символы и синтаксические конструкции. К числу основных управляющих символов и синтаксических конструкций относятся:

{,} – скобки открытия и закрытия подуровней;

! – комментарий (символы после этого знака не вставляются в результирующий текст);

@ – перевод строки (если в лексеме будет стоять два таких знака, то в тексте будет выполнено два перевода строки и т.д.);

< ? > – вставка произвольной строки текста (без перехода в редактор результирующего текста);

-- – вставка произвольных символов (по числу символов подчеркивания);

| ; ; | – выбор одной из перечисленных альтернатив;

[,] – скобки открытия и закрытия текстового блока;

Подуровни (поддеревья) и текстовые блоки могут быть именованными. Соответственно, вызов именованных поддеревьев и блоков может осуществляться по их имени, например: {одышка}, [рецепт] и т.д. В качестве одного из ограничений  $S()$  выступает следующее: каждая открытая скобка «{» (или «[») должна быть закрыта скобкой «}» (или «]»). Данное ограничение проверяется, например, на этапе преобразования текстового представления ЛД в машинное.

Если каждый кортеж  $W_i$  расположить с новой строки, то в качестве значений  $a_i$  может использоваться число пробелов от начала строки до первого символа словоформы  $L_i$ . Ниже представлены примеры лексических деревьев:

#### ЖАЛОБЫ НА

```
|постоянную; приступообразную| !головную боль {
  |умеренную; интенсивную|
  головную боль
  |возникающую; усиливающуюся| при
  повышении давления
  физической нагрузке
  перемене положения |головы; тела|
  < ? >
  в течение ____ |лет; года; год; мес; дн| }
|постоянное; приступообразное| !головокружение {головокружение}
```

#### НАЗНАЧЕНИЯ {

```
@- лизенил |0,025 мг;0,2 мг форте| по |1/4;1/2;1;1,5;2| таб |1;2;3| [
  раза в день в непрерывном режиме под контролем уровня пролактинина в крови. Контроль в __ мес.]
...}
```

С помощью чисел  $a_i$  или отступов лексем от левого края можно автоматически управлять расстановкой знаков препинания. Так, если несколько последовательных лексем имеют один и тот же отступ (больше нуля), то при выборе этих лексем программа будет пытаться поставить запятую (если нет другого знака). Если лексема начинается с заглавной буквы и имеет нулевой отступ, то в конце предыдущей фразы программа будет пытаться поставить точку.

Формально алгоритм автоматической расстановки знаков препинания может быть описан следующим образом. В процессе работы с ОПЯ автоматически формируется числовой массив  $M_k = \langle b_1, b_2, \dots, b_k \rangle$ . При выборе очередного кортежа  $W_i$  выполняются следующие действия:

Если  $a_i = 0$ , то в конец предыдущего текста ставится точка (если нет других знаков препинания), а затем добавляется выбранная лексема.

Иначе, если  $a_i > b_k$ , то  $b_{k+1} := a_i$ ,  $M_k \rightarrow M_{k+1}$  и выбранная лексема добавляется в результирующий текст без какого-либо знака препинания;

Иначе, если  $a_i = b_j$ , где  $j < k$ , то  $M_k \rightarrow M_j$  и в конец предыдущего текста ставится запятая (если нет других знаков), а затем добавляется выбранная лексема.

Иначе, если  $b_{j-1} < a_i < b_j$ , то  $b_j := a_i$ ,  $M_k \rightarrow M_j$  и выбранная лексема добавляется в результирующий текст без какого-либо знака препинания.

Приведенные выше синтаксические конструкции ОПЯ позволяют легко решить проблемы приставок, окончаний, отрицаний, заглавных букв. Так, возможна запись:

|Н;н|аличие уплотнени|я;й| |без;|болезненн|ого;ых| ...

Эта же словоформа может быть представлена следующим фрагментом ЛД:

|Н;н|аличие  
 уплотнения  
 уплотнений  
 безболезненного  
 безболезненных  
 болезненного  
 болезненных  
 ...

Выбор той или иной формы записи ЛД остается за конечным пользователем. Редактирование ЛД<sub>t</sub> осуществляется любым текстовым редактором (под DOS).

Машинное представление лексического дерева (индекс m) можно представить в виде упорядоченного множества трехэлементных кортежей:

$ЛД_m = \{ \langle a_i, \gamma_i, L'_i \rangle \mid a_i \in N, \gamma_i - \text{кортеж управляющих символов, } L'_i - \text{словоформа, } i = 1 \dots n \}$ .

При преобразовании текстового представления ЛД в машинное скобки { } и [ ], обозначающие подуровни и блоки, преобразуются в порядковый номер подуровня и удаляются из L ( $L \rightarrow L'$ ). Сам порядковый номер подуровня помещается в  $\gamma_i$ . Туда же помещается

признак «подуровень / блок», а также признак «стандартная лексема». Все стандартные лексемы могут быть выбраны нажатием одной клавиши. В процессе преобразования происходит также подстановка всех поименованных фрагментов ЛД в те места, где они вызывались. При этом теряется информация о поименованных фрагментах ЛД<sub>t</sub>. С помощью лексического процессора можно непосредственно в ЛД<sub>m</sub> вставлять, удалять и модифицировать лексемы, минуя текстовое представление ЛД. При обратном преобразовании ЛД<sub>m</sub> → ЛД<sub>t</sub> порядковый номер подуровня и признак «подуровень / блок» преобразуются соответственно в скобки { } и [ ]. Если ЛД<sub>t</sub> содержало поименованные блоки, то при обратном преобразовании подобная структура не воспроизводится.

ЛД<sub>m</sub> хранится в таблице базы данных следующего вида (атрибуты):

Имя поля	Тип поля	Назначение
FNAM	C(40)	Название лексического дерева
SCEN	M	Собственно лексическое дерево
CODE	C(4)	Код лексического дерева
PREV	C(4)	Код предыдущего уровня
NEXT	C(1)	Указатель наличия подуровня

Как видно из представленной структуры в одной таблице может храниться любое количество ЛД. Самых таблиц также может быть сколь угодно много. Как правило, для каждого шаблона медицинского документа создается своя таблица БД, содержащая ЛД, связанные с этим документом. Например, в информационной системе Днепропетровского областного диагностического центра насчитывается более 800 таблиц с ЛД.

Визуальное представление лексического дерева (ЛД<sub>v</sub>) имеет вид классического древовидного списка (TreeView – компонента). Навигацию по дереву, выбор и обработку лексем, а также формирование результирующего текста осуществляет лексический процессор.

## 2. Семантический процессор

В основе работы семантического процессора (СП) лежит понятие «семантическая операция». В общем случае, в качестве семантических операций могут выступать: выборка необходимых данных из БД; занесение данных в БД; логические и арифметические операции с данными. В настоящей работе ограничимся рассмотрением последнего вида семантических операций.

Метаинформация, содержащая описание шаблонов медицинских документов (форм), представляет собой справочник шаблонов, а также набор таблиц специальной структуры, описывающих каждый шаблон в отдельности. Любая запись в такой таблице содержит описание некоторого поля в документе: тип поля, расположение, маска ввода, способ ввода (справочник, лексика, редактируемое, вычисляемое), границы нормы, указатель на «Расчет», указатель на вид семантической операции, идентификатор, наименование файла справочника, лексики, семантики. Если в каком-либо поле стоит указатель на «Расчет», то запускается выполнение определенной семантической операции. Программная реализация семантических операций разных видов может существенно различаться и постоянно совершенствуется. Один из видов семантических операций реализован с помощью скриптов. Скрипт представляет собой набор текстовых строк на формальном языке, напоминающем Паскаль. Как и Паскаль, язык требует явного описания всех используемых переменных. Выполняет скрипт специализированный интерпретатор. Например, в документе «Эхокардиография» расчет параметра «Частота Сердечных Сокращений» может выполняться после ввода параметра «Интервал RR», с помощью следующего скрипта:

```
var
    hss:real {частота сердечных сокращений}
    RR:real {интервал RR}
endvar
hss:=60/RR {расчет переменных}
write("hss:"hss) {возврат значения}
```

Программный скрипт для определения параметра «Сократительная способность миокарда» может выполняться после ввода параметра «Фракция выброса»:



```
var
  FB:real {фракция выброса}
  s_s_m:string { Сократительная способность миокарда }
endvar
if (FB < 45) then
  s_s_m:='снижена'
else
  s_s_m:='удовлетворительная'
writeln('s_s_m:'s_s_m) {возврат значения}
```

Для хранения программных скриптов используется отдельный файл (таблица баз данных), который связан с шаблоном документа. Поля таблицы базы данных для хранения скриптов: 1) идентификатор – должен совпадать с идентификатором в шаблоне документа; 2) комментарий – строка, описывающая назначение идентификатора; 3) программный скрипт. При расчете некоторых параметров СП использует такие данные о пациенте как: пол, рост, вес, возраст. Эти параметры объявляются как глобальные переменные и передаются СП при необходимости.

Работа СП осуществляется следующим образом. При открытии шаблона документа выполняется проверка наличия файла со скриптами. Если файл найден, то выполняется подключение СП (загрузка dll-библиотеки интерпретатора и файла со скриптами). При вводе данных в шаблон выполняется проверка на наличие идентификатора для текущего поля ввода. Если он есть, то выполняется проверка наличия программного скрипта для этого идентификатора (в файле скриптов). Анализируются переменные, используемые при вычислениях, и при необходимости выполняется передача параметров, ранее введенных в документ. Далее, программный скрипт передается интерпретатору для исполнения. Результат работы интерпретатора возвращается в промежуточный буфер обмена. В буфер помещается идентификатор и его значение. По идентификатору выполняется поиск поля с таким именем в шаблоне документа. При нахождении результат из буфера заносится в поле ввода шаблона. Для отображения данных в поле может использоваться маска ввода, в этом случае выполняется преобразование значения результата для отображения по маске.

## Список литературы

- [1] Алпатов А., Прокопчук Ю. Автоматизированная информационная система участковой поликлиники // Укр. журн. мед. техники и технологии. 1994. №1–2. С. 68–72.
- [2] Алпатов А., Пиляев В., Прокопчук Ю. Автоматизированная информационная система медсанчасти Никопольского завода ферросплавов // Информационные технологии и программно-аппаратные средства в медицине, биологии и экологии. Ч. I. Киев: Мединформ, 1998. С. 61–66.
- [3] Алпатов А., Прокопчук Ю., Пилипченко И. Лексический процессор медицинского назначения // Укр. журн. мед. техники и технологии. 1998. №4. С. 57–58.
- [4] Ovsuk L., Prokopchuk Y., Pilipchenko I., Kornilova S. Automated working place of the mammolog service of diagnostic centre // School of Fundamental Medicine Journal. 1998. V. 4. No. 2. P. 94.
- [5] Alpatov A., Prokopchuk Y. Techno-economic analysis of efficiency uses of cosmic technologies in medicine // School of Fundamental Medicine Journal. 1998. V. 4. No. 2. P. 93.
- [6] Алпатов А., Прокопчук Ю. Алгоритм прогнозирования заболеваний на основе активных баз данных // Информационные базы данных в медицине. Мониторинг здоровья населения и окружающей Среды. (Египет – Хургада, 1999). М., 1999. С. 11.
- [7] Прокопчук Ю.А., Овсяк Л.И., Пилипченко И.А. Прогнозирование угрожающих состояний на основе активных баз данных // Вестник ХГПУ «Системный анализ, управление и информационные технологии». Вып. 72. 1999. С. 193–198.
- [8] Алпатов А., Горбаненко С., Каминская Е., Прокопчук Ю., Харченко В., Шкрям Я. Опыт разработки и эксплуатации информационной системы участковой поликлиники // Укр. журн. мед. техники и технологии. 1999. №2–3. С. 112–119.
- [9] Алпатов А.П., Прокопчук Ю.А., Костра В.В., Пилипченко И.А. Интеллектуальные управляющие информационные системы // Сб. трудов межд. конф. по автоматическому управлению «Ав-

томатика – 2000». Т. 6. Львов: ГосНИИ информационной инфраструктуры, 2000. С. 20–24.

- [10] Прокопчук Ю.А., Алпатов А.П., Огданский Н.Ф., Харченко О.А., Костра В.В. Интеллектуальные модули информационных систем // Сб. трудов межд. конф. «Искусственный интеллект – 2000» (п. Кацевели). Донецк: Ин-т проблем искусств. интеллекта, 2000. С. 315–317.
- [11] Харченко О.А., Прокопчук Ю.А., Костра В.В., Коноваленко М.Е. Информационная система диагностического отделения медицинского НИИ // Сб. Трудов филиала МГТУ им. Н.Э. Баумана в г. Калуге. Специальный выпуск. Материалы межд. конф. «Приборостроение – 2000» (п. Симеиз). Калуга: Изд-во Н. Бочкаревой, 2000. С. 349–352.
- [12] Алпатов А.П., Прокопчук Ю.А., Костра В.В. Семантический процессор для медицинских приложений // Сб. Трудов филиала МГТУ им. Н.Э. Баумана в г. Калуге. Специальный выпуск. Материалы межд. конф. «Приборостроение – 2000» (п. Симеиз). Калуга: Изд-во Н. Бочкаревой, 2000. С. 353–357.
- [13] Прокопчук Ю.А., Харченко О.А., Костра В.В. Интеллектуальный интерфейс медицинских информационных систем // Сб. трудов VIII междунар. форума «Информационные технологии и интеллектуальное обеспечение в здравоохранении и охране окружающей среды» (Турция, 2000 г.). М., 2000. С. 47.