

ОБРАБОТКА ЗАПРОСОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

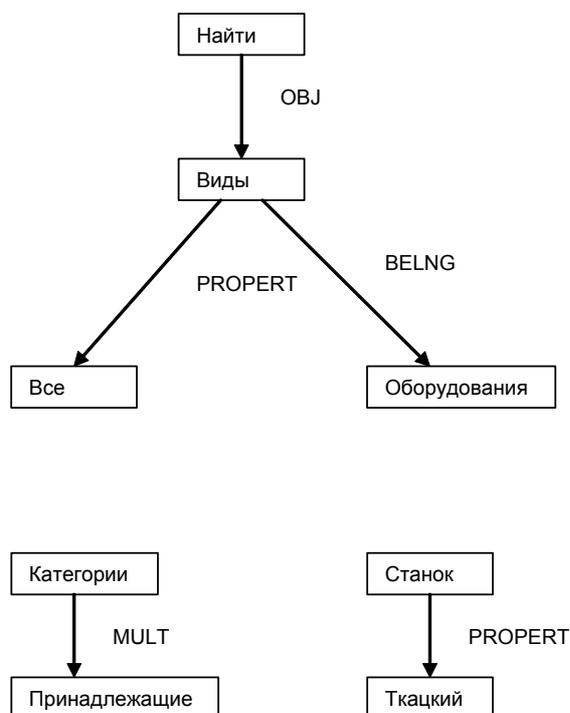
МАЛАФЕЕВА Т.Н., асп., ЗУБКОВ В.П., канд. техн. наук, ШУЛЬПИН А.А., канд. техн. наук

Рассматриваются вопросы использования созданных средств проведения морфологического, синтаксического и семантического анализов относительно запросов на естественном языке в области энергосбережения в текстильной промышленности

Преобразование запроса с естественного языка в SQL-запрос является актуальной задачей. Группой отделения интеллектуальных систем института лингвистики в 1999-2003 гг. был создан синтаксический анализатор, который дополнен возможностью проведения семантического анализа на уровне толковых словарей типа Ожегова или Даля. Разработаны динамически загружаемые библиотеки (DLL), которые можно подключать для проведения морфологического, синтаксического и неглубокого семантического анализа. В результате опытной эксплуатации разработанных динамически загружаемых библиотек было выявлено следующее: результаты синтаксического анализа не всегда однозначны; результаты семантического анализа часто выдают «лес», а не «дерево». На рисунке показан результат испытания запроса на естественном языке «Найти все виды оборудования, принадлежащие категории «Ткацкий станок»». Как видно из рисунка, в результате семантического анализа получается «лес», а не «дерево» для одного предложения. В принципе, даже для нескольких предложений, объединенных общим смыслом, должно получаться «дерево». Причем на этом рисунке получается семантический граф первого уровня (по терминологии интеллектуальной системы автоматизированного программирования СемС31). В виде помеченных списков семантическая сеть может быть представлена следующим образом:

```
(СемС31_1 (НВ (ЗН Найти))
           (КВ (ЗН Виды))
           (ОТН (ЗН OBJ)))
(СемС31_2 (НВ (ЗН Виды))
           (КВ (ЗН Все))
           (ОТН (ЗН PROPERT)))
(СемС31_3 (НВ (ЗН Виды))
           (КВ (ЗН Оборудования))
           (ОТН (ЗН BELNG)))
(СемС31_4 (НВ (ЗН Категории))
           (КВ (ЗН Принадлежащие))
           (ОТН (ЗН MULT)))
(СемС31_5 (НВ (ЗН Станок))
           (КВ (ЗН Ткацкий))
           (ОТН (ЗН PROPERT)))
```

Для получения связного графа в виде дерева необходимо связать вершину «Виды» с вершиной «Категории» и вершину «Категории» с вершиной «Станок». Кроме этого, такие вершины, как «Оборудования», «Категории», «Принадлежащие», хранят неоднозначную морфологическую информацию, что в списочном виде будет выглядеть следующим образом:



Семантический граф зпроса

```
(СемС31_1 (НВ (ЗН Найти))
           (КВ (ЗН Виды))
           (ОТН (ЗН OBJ)))
(СемС31_2 (НВ (ЗН Виды))
           (КВ (ЗН Все))
           (ОТН (ЗН PROPERT)))
(СемС31_3 (НВ (ЗН Виды))
           (КВ (ЗН Оборудования))
           (ОТН (ЗН BELNG)))
(СемС31_4 (НВ (ЗН Категории))
           (КВ (ЗН Принадлежащие))
           (ОТН (ЗН MULT)))
(СемС31_5 (НВ (ЗН Станок))
           (КВ (ЗН Ткацкий))
           (ОТН (ЗН PROPERT)))
(СемС31_6 (НВ (ЗН Виды))
           (КВ (ЗН Категории))
           (ОТН (ЗН BELNG)))
(СемС31_7 (НВ (ЗН Категории))
           (КВ (ЗН Станок))
           (ОТН (ЗН BELNG)))
```

Возникает проблема разработки надстройки над созданными модулями морфологического, синтаксического и семантического анализов относительно конкретной прикладной области. Так

как в заданном запросе мы находимся всегда в конкретных условиях относительно предметной области, то проблема решается.

Алгоритм расширения возможностей разработанных программ морфологического, синтаксического и семантического анализов следующий:

- A1) Открыть файл Ф1, содержащий морфологическую, синтаксическую и семантическую информацию.
- A2) Открыть файл Ф2 для записи окончательной морфологической информации.
- A3) Читать информацию об очередной словоформе из файла Ф1.
- A4) Если файл закончился, то перейти на А8.
- A5) Если морфологическая информация о словоформе неоднозначная, то вызвать модуль удаления неоднозначной морфологической информации.
- A6) Переписать всю существующую информацию с однозначной морфологической информацией в файл Ф2.
- A7) Перейти на А4.
- A8) Закрыть файлы Ф1 и Ф2.
- A9) Открыть файлы Ф2 и Ф3 для записи окончательной информации.
- A10) Читать информацию об очередной словоформе С1 из файла Ф2.

A11) Читать информацию о следующей словоформе С2 из файла Ф2.

A12) Если файл закончился, то перейти на А20.

A13) Если словоформы С1 и С2 связаны, то записать информацию из С1 в файл Ф3 и перейти на А14, в противном случае перейти на А15.

A14) Переслать информацию, содержащуюся в С2, в С1 и перейти на А11.

A15) Читать информацию о следующей словоформе в С3 из файла Ф2.

A16) Если файл закончился, то перейти на А19.

A17) Если словоформы С1 и С3 связаны, то записать информацию из С1 в файл Ф3, переслать информацию, содержащуюся в С3, в С1 и перейти на А15.

A18) Перейти на А15.

A19) Вызвать модуль, который может связать С1 и С2. Записать в файл Ф3 информацию из С1. Переслать С2 в С1, перейти на А11.

A20) Закрыть файлы Ф2 и Ф3.

После работы данного алгоритма в файле Ф3 будет создано одно «дерево», по которому можно строить SQL-запрос. Для построения SQL-запроса необходимы справочники и правила в базах знаний, согласно которым возможно конструктивное построение запросов. Алгоритм построения SQL-запросов будет рассмотрен в следующей статье.