

УДК 004.89 + 004.5

ББК 32.813

## **ОЦЕНКА ЭФФЕКТИВНОСТИ ЛОГИЧЕСКОГО АНАЛИЗА ЕСТЕСТВЕННЫХ ЯЗЫКОВ В СИСТЕМАХ УПРАВЛЕНИЯ СЛОЖНЫМИ ТЕХНИЧЕСКИМИ ОБЪЕКТАМИ <sup>1</sup>**

**Крайванова В.А. <sup>2</sup>**

*(Алтайский государственный технический университет им.  
И.И. Ползунова, г. Барнаул)*

*В статье рассмотрена проблема построения системы семантического анализа текстов на естественном языке на основе лишь синтаксических знаний о языке. Предложена математическая модель и показана ее эффективность. На основе построенной модели рассмотрена архитектура интеллектуального управления техническими объектами на примере систем „Умный дом“.*

Ключевые слова: естественно-языковой интерфейс, человеко-машинное взаимодействие, искусственный интеллект.

### **Введение**

Задача создания интерфейса для общения с компьютером на естественном человеческом языке (ЕЯ) не нова, однако сейчас она вновь обретает остроту и актуальность в свете особенностей современных компьютерных систем. Это связано, прежде всего, со следующими тремя факторами. Во-первых, на сегодняшний день сфера применения компьютеров настолько широка и разнообразна, что использовать для всех задач универсальный интерфейс человеко-машинного взаимодействия неэффективно. Каж-

---

<sup>1</sup> Автор признателен профессору Е.Н. Крючковой за ценное обсуждение содержания статьи.

<sup>2</sup> Варвара Андреевна Крайванова, аспирант ([krayvanova@yandex.ru](mailto:krayvanova@yandex.ru)).

дый вид интерфейса становится всё более специализированным. Во-вторых, нарастание сложности технических систем требует от субъекта управления всё больших усилий по запоминанию сценариев взаимодействия: комбинаций действий, нажатий кнопок, голосовых команд. В-третьих, пользователю приходится приспосабливаться к формальному компьютерному интерфейсу. Альтернативой является построение человеко-машинного взаимодействия на основе естественного человеческого языка, но данная задача необычайно сложна с точки зрения формализации. Задача синтаксического разбора фраз ЕЯ в определенной степени уже решена, чего нельзя сказать о смысловом анализе текстов.

Одним из прикладных вопросов в этой области является перевод задания, сформулированного в фразах ЕЯ, в формальные команды исполнителя-компьютера. Вопрос имеет широкую сферу применения: от построения естественно-языковых интерфейсов к базам данных (например, отечественный коммерческий проект InBase [1]) до сред программирования сверхвысокого уровня.

На практике, как правило, для каждой такой задачи строится своя семантическая модель. Такие модели сложны, чрезвычайная многозначность ЕЯ на вычислительном уровне приводит к комбинаторному взрыву [2].

Тем не менее, далеко не каждая задача требует глубокого семантического анализа текста, учитывающего структуру дискурса. Во многих прикладных областях буквальное понимание текстов на уровне простейшей логики было бы достаточным. Примером такой области служат системы „Умных домов“ [3].

**Цель работы** - построение системы семантического анализа текстов на ЕЯ, опирающейся лишь на синтаксические знания о языке.

## **1. Постановка задачи**

Умный дом - это общий термин, используемый для описания многочисленных продуктов и технологий, которые значительно расширяют функциональность нынешних домашних устройств и делают это, как правило, на базе конвергированной сетевой ин-

фраструктуры. Сетевые аудио-, видео- и телефонные службы, а также служба передачи данных применяются не только в офисах, но и в домах. Технологии умного дома позволяют разнообразным устройствам использовать сети передачи данных и Интернет для выполнения таких задач, которые не по плечу обычным бытовым приборам, например, каждый день включать свет в шесть утра, выключать систему полива газона, когда идет дождь, и в определенное время перенаправлять сигнал DVD-проигрывателя с телевизора в гостиной на телевизор в спальне. Набор уже доступных приложений и устройств (для оборудования умного дома) расширяется с каждым месяцем. Сравнительно недавно появившиеся в России интеллектуальные системы управления домом открывают широкие возможности для создания безупречно комфортного жилого пространства. В доме автоматизируются и объединяются в единую сеть системы видеонаблюдения, отопления, освещения, домашние кинотеатры, компьютеры, акустика и так далее. Способы, которыми пользователь может управлять „Умным домом“ [3]:

- кнопочные панели;
- сенсорные панели;
- управление компьютером;
- управление с КПК;
- управление с телефона;
- радиоуправление;
- голосовое управление;
- управление через Интернет.

Все системы управления „Умными домами“ являются строго формальными. Пользователю приходится вспоминать, какую кнопку надо нажать, чтобы выключить свет или какую фразу произнести, чтобы открыть двери гаража. Сформулировать свое желание на естественном языке для человека гораздо проще, чем на

формальном, а чтобы понять это желание в контексте управления „Умным домом“ достаточно простейшей логики. Таким образом, задачей работы является построение программного робота, способного преобразовывать пожелания пользователя, выраженные в виде фраз на ЕЯ, в команды на формальном языке управления „Умным домом“.

## 2. Математическая модель

Для моделирования естественного языка необходимо задать следующие :

- лексикон как множество известных в модели слов;
- формальное представление фразы ЕЯ на основе ее синтаксической структуры;
- представление команд системы управления;
- механизм анализа фраз.

В качестве элемента лексикона возьмем синтаксически неделимую единицу – слово. Из множества  $W$  слов предметной области выделим множество  $N$  понятий естественного языка. Нечеткое **отношение обобщения**  $Gen : W \times N \rightarrow [0..1]$  определяет степень уверенности модели в том, что некоторое понятие  $p \in N$  обобщает слово  $w \in W$ .

Нечеткое **отношение синонимичности**  $Syn : W \times W \rightarrow [0..1]$  определяет степень уверенности системы в том, что некоторое слово  $w_A \in W$  является синонимом к слову  $w_B \in W$ .

Формально, фразу на естественном языке определим как функциональную форму, в которой в качестве функциональных символов выступают слова лексикона  $W$ :

- $w \in W$  - любое слово из лексикона модели является фразой;
- $\neg\phi$  - отрицание фразы является фразой;

- $w([r_1 : \phi_1], [r_2 : \phi_2], \dots [r_n : \phi_n]), n > 0, \phi_i$  - подчиненная фраза,  $r_i$  - тип аргумента,  $w \in W$  - слово с перечисленными в скобках подчиненными фразами.

Пусть  $Rel$  - конечное множество типов аргументов модели.

Функциональную форму фразы  $\phi$  удобно изображать в виде дерева, вершины которого помечены словами  $w \in W$ , а ребра - типами аргументов  $r \in Rel$  (Рис. 1). Корень дерева фразы  $\phi$  будем обозначать как  $Root(\phi)$ . Пусть  $u$  - узел помеченного дерева фразы. Тогда  $Word(u)$  - пометка узла  $u$ ,  $Relation(u)$  - пометка ребра, соединяющего узел  $u$  с родительским,  $Args(u)$  - множество дочерних узлов узла  $u$ .  $Negation(u)$  - признак отрицания узла  $u$ .

Получить такое помеченное дерево фразы из предложения на ЕЯ, можно, например, с помощью библиотек Dialing [4]. Для иллюстрации используются деревья, полученные этим анализатором.

Сегодня к нам придут дорогие гости

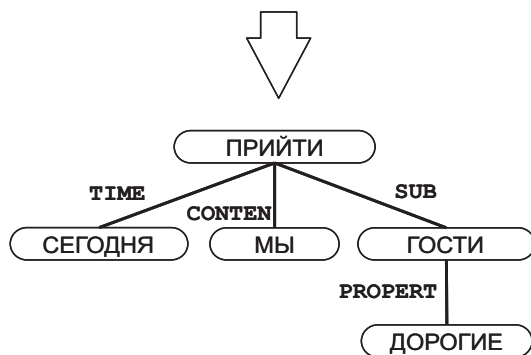


Рис. 1. Пример построения дерева фразы

Команда системы управления представляется парой:

$e = \langle \nu, Parameters \rangle$ ,

где  $\nu$  - функциональная форма, задающая определение команды на ЕЯ. Некоторые функциональные символы в  $\nu$  могут быть помечены не словами из  $W$ , а параметрами из множества

*Parameters.*  $Parameters = \{p | p = \langle name_i, N_i \rangle, N_i \subseteq N\}$  - множество параметров команды, где  $name_i$  - имя параметра,  $N$  - множество понятий.

В узлы дерева  $\nu$ , помеченные параметром  $name_i$ , может быть подставлено некоторое дерево  $\phi$ , такое, что  $\exists n : Gen(Word(Root(\phi)), n) > 0, n \in N_i$ . Пусть  $\tau(p_i)$  - уровень достоверности унификации параметра.

$$(1) \quad \tau(p_i) = Gen(Word(Root(\phi)), n)$$

Проверка наличия в некоторой фразе  $\phi$  команды  $e$  и вычисление параметров команды производится с помощью операции унификации.

$$(2) \quad \langle \gamma, Values \rangle = Unification(\phi, e)$$

Результатом унификации являются:

- уровень релевантности  $\gamma$ , который определяет степень уверенности системы в наличии команды.
- множество  $Values = \{\langle p_i, \phi_i \rangle\}$ , где  $p_i$  - параметры команды  $e$ ,  $\phi_i$  - значения параметров. Причем для каждого параметра  $p_i$  должно существовать единственное значение  $\phi_i$ .

Обозначим  $Value(p_i) = \phi_i$ .

Задача логического анализа заключается в том, чтобы свести некоторую начальную фразу  $\phi$  к множеству фраз, определяющих команды и понятных объекту управления. Механизм преобразования фраз описывается конечным множеством правил контекстной замены *Rules*. Введем три группы правил контекстной замены.

**Логическое правило** задается следующим образом:

$$r_L = \langle \nu \rightarrow \sigma, Parameters \rangle,$$

где  $\nu$  и  $\sigma$  - гипотеза и следствие правила, представленные двумя суперпозициями, некоторые функциональные символы которых, возможно, помечены не словами ЕЯ, а параметрами из

множества *Parameters*. *Parameters* - множество параметров правила, аналогичное множеству параметров команды.

**Правило-определение** используется для перевода функциональных форм устойчивых словосочетаний в слова лексикона. Правило-определение задается следующим образом:

$$r_O = \langle \nu \rightarrow c, Parameters \rangle,$$

где  $\nu$  - словосочетание в функциональной форме, содержащее, возможно, параметры;  $c$  - определяемое слово из лексикона  $W$ , *Parameters* - множество параметров.

**Заключительное правило** представляет собой шаблон команды для объекта управления или фразу-ответ на ЕЯ. Заключительное правило задается парой

$$r_E = \langle \nu, Parameters \rangle,$$

где  $\nu$  - функциональная форма, представляющая шаблон команды объекта управления или фразу-ответ на ЕЯ.

**Цепочка вывода** из функциональной формы  $\phi$  - это конечная последовательность функциональных форм  $\langle \phi_0, \phi_1, \dots, \phi_n \rangle$ , таких, что  $\phi_0 = \phi$ , форма  $\phi_{i+1}$  непосредственно выводима из  $\phi_i$  для  $0 < i < n - 1$  с помощью некоторого правила  $r_i \in Rules$ , форма  $\phi_n$  получена из формы  $\phi_{n-1}$  с помощью заключительного правила  $r_n \in Rules$ .

На основе введенных определений задачу логического анализа фразы  $\phi$  можно сформулировать следующим образом: найти множество  $E_\phi$  всех заключительных функциональных форм, выводимых из данной функциональной формы  $\phi$  с помощью цепочек вывода длины не более *Lenght*. Очевидно, что при такой постановке задачи множество  $E_\phi$  конечно.

Если рассматривать данную модель с точки зрения практической применимости, возникает проблема, типичная для интеллектуальных систем: высокая временная сложность алгоритмов. Предлагаемый способ решения этой проблемы рассмотрен далее.

### 3. Задача оптимизации

В большинстве случаев фраза на ЕЯ характеризуется уникальным упорядоченным набором слов - пометок вершин. Схемы

построения фраз, задаваемые пометками ребер, определяются не обширным лексиконом, а структурной спецификой выбранного языка, и потому имеют тенденцию к повторению. Эта особенность ЕЯ отражается, прежде всего, в синтаксисе. Учет этого обстоятельства позволяет, как показано далее, значительно снизить временную сложность алгоритмов предложенной модели.

Схема  $Template(\phi)$  - это дерево фразы  $\phi$  без пометок вершин. Для хранения схем гипотез всех правил базы и целевых шаблонов команд нами разработана специальная структура данных - дерево схем (ДС). Ребра ДС помечены типами отношений, возможными между лексемами во фразах на анализируемом языке. Каждый узел может быть связан данным типом отношения с не более чем одним потомком. Таким образом, каждый узел  $u_t$  ДС однозначно задается упорядоченным набором пометок ребер  $L(u_t)$  вдоль пути от корня дерева к узлу  $u_t$ . Каждый узел  $u_t$  содержит список правил, в гипотезе которой существует узел  $u$ , такой что  $L(u) = L(u_t)$ .

Заметим, что производительность ДС зависит не от количества внесенных в него фраз, а от количества узлов в этих фразах. Эксперименты, результаты которых представлены ниже, показали эффективность ДС не только как структуры организации базы правил, но и как хранилища уже полученных фраз для предотвращения повторного вывода.

#### **4. Результаты тестирования производительности**

Были проведены вычислительные эксперименты, позволяющие проанализировать эффективность разработанных алгоритмов. В экспериментах измерялось только „чистое“ время работы системы, не включающее время перевода фраз на ЕЯ во внутреннее представление.

Поведение системы значительно зависит от конкретного содержания предложений. Это связано с наличием экстенсивных правил. Пусть дерево фразы  $s_1$  - результат применения правила  $r$  к дереву фразы  $s$ . Правило  $r$  назовем экстенсивным, если оно применимо к  $s_1$ . Такие правила вызывают порождение сильно вет-

вляющихся деревьев вывода с большой высотой, что существенно влияет на производительность системы. Для более объективной оценки времени работы системы каждая точка графика получается как среднее значение времени работы пяти тестов. В качестве входного текста для каждого из пяти тестов использовалась случайная выборка предложений требуемого размера. К каждому предложению применимо хотя бы одно правило из базы правил. Для алгоритмов выбора правила ЛВП и ДС в точках с одинаковым количеством правил анализируются одинаковые тексты.

### **1. Сравнение алгоритмов выбора правила ЛВП и ДС. Зависимость от размера текста.**

Условия эксперимента:

- База правил - 200 правил.
- Структура данных для хранения полученных результатов - красно-черное дерево.
- Максимальная глубина дерева вывода - 5 шагов.
- Диапазон изменения размера текста: от 10 до 120 предложений; шаг: 10 предложений.

На рисунке 2 изображена полученная зависимость. Пунктирными линиями обозначены коридоры значений.

### **2. Сравнение алгоритмов выбора правила ЛВП и ДС. Зависимость от количества правил.**

Условия эксперимента:

- Диапазон изменения размера базы правил - от 15 до 390 правил; шаг: 15 правил.
- Структура данных для хранения полученных результатов - красно-черное дерево.
- Максимальная глубина дерева вывода - 5 шагов.
- Размер текста: 128 предложений.

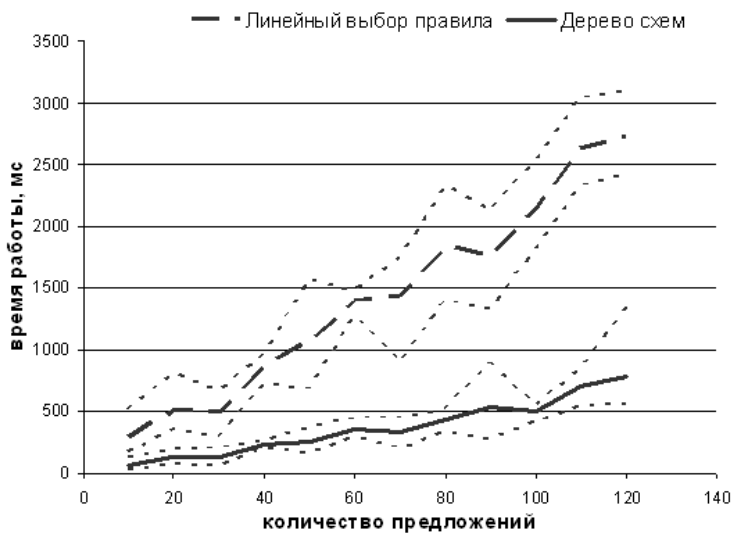


Рис. 2. Сравнение алгоритмов выбора правила ЛВП и ДС.  
Зависимость от размера текста.

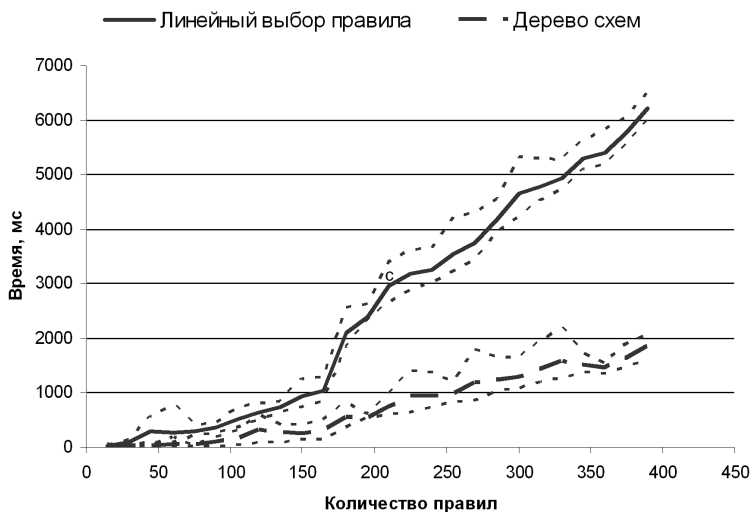


Рис. 3. Сравнение алгоритмов выбора правила ЛВП и ДС.  
Зависимость от количества правил.

На рисунке 3 изображена полученная зависимость. Пунктирными линиями обозначены коридоры значений.

Серии 1 и 2 экспериментов демонстрируют несомненное превосходство алгоритма ДС в сравнении с алгоритмом ЛВП для выбора применяемого правила.

**3. Сравнение ДС и красно-черного дерева как структур данных для хранения полученных результатов (для предотвращения повторного вывода) для различных способов выбора правила: ЛВП и ДС.**

Условия эксперимента:

- База правил - 200 правил.
- Максимальная глубина дерева вывода - 7 шагов.
- Диапазон изменения размера текста: от 10 до 120 предложений; шаг: 10 предложений.

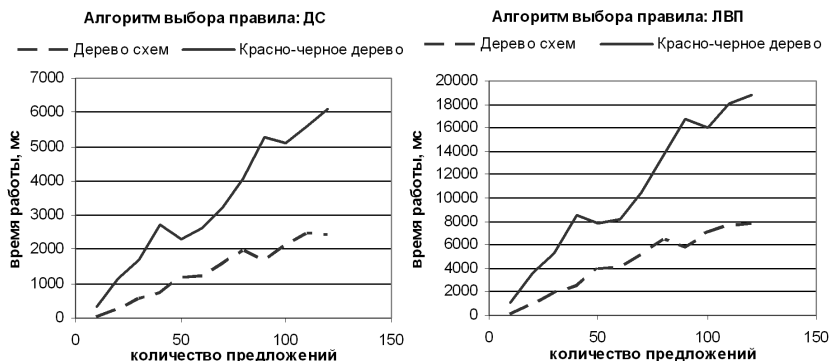


Рис. 4. Сравнение ДС и красно-черного дерева как структур данных для хранения полученных результатов (для предотвращения повторного вывода) для различных способов выбора правила: ЛВП и ДС.

Эксперимент показал, что в качестве структуры данных для хранения полученных результатов ДС значительно эффективнее красно-черного дерева. Однако это отличие заметно лишь при

большой глубине дерева логического вывода. В проведенном эксперименте алгоритм логического вывода получил порядка 1400 различных фраз для текста из 120 предложений. Если ограничить максимальную глубину дерева вывода 5 шагами, то количество получаемых различных фраз - около 600, и различия в производительности между красно-черным деревом и ДС незаметны.

#### 4. Зависимость времени работы алгоритмов от глубины вывода для различных способов выбора правила: ЛВП и ДС.

Условия эксперимента:

- База правил - 200 правил.
- Структура данных для хранения полученных результатов - красно-черное дерево.
- Диапазон ограничения на максимальную глубину дерева вывода - от 3 до 7 шагов.
- Диапазон изменения размера текста: от 10 до 120 предложений; шаг: 10 предложений.

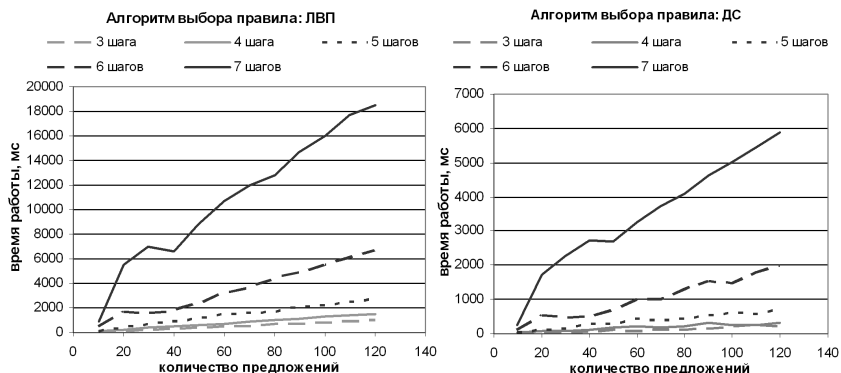


Рис. 5. Зависимость времени работы алгоритмов от глубины вывода для различных способов выбора правила: ЛВП и ДС.

Эта серия экспериментов показывает, что производительность системы при росте ограничения на максимальную глубину дерева вывода очень быстро падает (рис. 5).

## **5. Общая архитектура системы**

В качестве языка программирования для реализации предложенной математической модели выбран мультиплатформенный язык Java. Взаимодействие модулей реализации системы представлено на рисунке 6[5].

Модуль первичного семантического анализа преобразует фразы на естественном языке во внутреннее представление системы - функциональные формы. Первичный семантический анализ использует внешний синтаксический анализатор текстов на естественном языке. Модуль первичного семантического анализа легко заменяем, что позволяет подключать к системе синтаксические анализаторы различных языков. В текущей реализации для синтаксического анализа используется продукт Dialing.

Задача модуля логического вывода - построить список команд для объекта управления из исходных гипотез путем построения цепочек логического вывода. Алгоритм логического вывода конфигурируется набором параметров, таких, как глубина дерева логического вывода и параметры обучения.

Преобразование результатов логического вывода в команды для „Умного дома“ производится в модуле интерпретации. В простейшем случае каждому шаблону команды соответствует мини-программа определенной системы „Умного дома“. Этот модуль является заменяемым и определяется программным интерфейсом объекта управления.

Через модуль графического интерфейса осуществляется взаимодействие с пользователем системы: ввод команд на естественном языке, интерактивное обучение системы с учителем, конфигурация и настройка системы.

Модуль инициализирующего обучения автоматизирует процесс построения базы знаний системы.

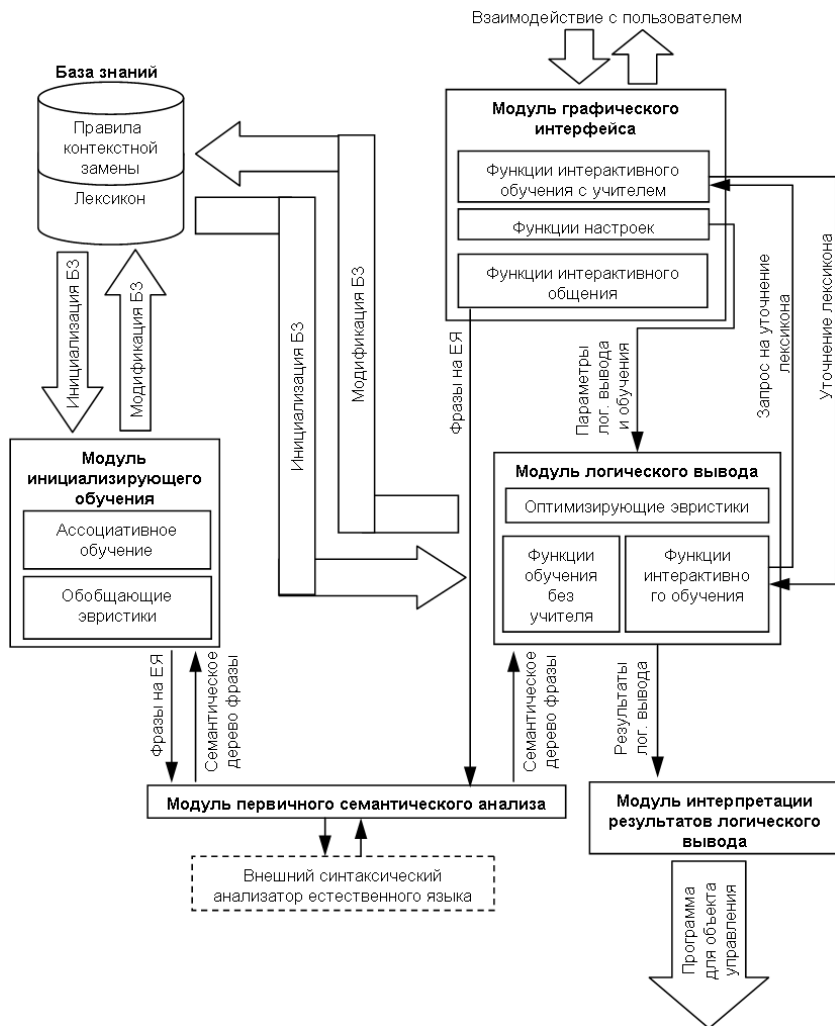


Рис. 6. Схема потоков данных системы.

## **6. Выводы**

Вычислительные эксперименты демонстрируют, что предложенная математическая модель адекватна моделируемому объекту в пределах поставленной задачи. Тестирование производительности показало высокую эффективность механизмов оптимизации. Модель представляет практический интерес в области интеллектуального управления техническими системами.

Использование разработанной программной библиотеки в комплексах „Умный дом“ позволит расширить функциональные возможности комплексов (возможно управление любыми электрическими бытовыми приборами без усложнения системы управления). Кроме того, предложенный механизм не требует от пользователя специальной подготовки и опыта работы с формализованными знаковыми системами.

## **Литература**

1. *Сайт Российского НИИ искусственного интеллекта.* Режим доступа: <http://artint.ru>
2. ЛЮГЕР Д. Ф. *Искусственный интеллект: стратегии и методы решения сложных проблем.* 4-е издание. - М.: Вильямс, 2003.
3. *Сайт компании House Control.* Режим доступа: <http://www.housecontrol.ru>
4. СОКИРКО А. В. *Реализация первичного семантического анализа в системе Диалинг.* Тр. междунар. семинара Диалог'2000 по компьютерной лингвистике и её приложениям, - Протвино, 2000
5. КРАЙВАНОВА В. А. *Система интеллектуального управления сложными адаптирующимися объектами.* Научная сессия ТУСУР-2008: Материалы докладов Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых. Томск, 5-8 мая 2008г.: в пяти частях. Ч. 2, Секция 9. Вычислительный интеллект. - Томск: В-Спектр, 2008. - стр. 78-81.

## **EFFICIENCY EVALUATION OF THE NATURAL LANGUAGE LOGIC ANALYSIS IN CONTROL SYSTEMS OF COMPLEX TECHNICAL OBJECTS**

**Varvara Krayvanova**, Altai State Technical University, Barnaul,  
post-graduate student(krayvanova@yandex.ru).

*Abstract: The article consider the problem of construction of system of the natural language texts semantic analysis, which base on only syntactic knowledge of the language. The mathematical model is offered and its efficiency is shown. The architecture of technical objects intellectual control is considered by the example of systems „Clever house“.*

**Keywords:** natural language interface, human-computer interaction, artificial intelligence.