

УДК 004.89 + 004.5

ББК 32.813

МОДЕЛЬ ЕСТЕСТВЕННО-ЯЗЫКОВОГО ИНТЕРФЕЙСА ДЛЯ СИСТЕМ УПРАВЛЕНИЯ СЛОЖНЫМИ ТЕХНИЧЕСКИМИ ОБЪЕКТАМИ И ОЦЕНКА ЭФФЕКТИВНОСТИ АЛГОРИТМОВ НА ЕЕ ОСНОВЕ ¹

Крайванова В.А. ²

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

В статье рассмотрена проблема построения интеллектуально-го человеко-машинного интерфейса на основе логического анализа фраз на естественном языке. Предложена математическая модель интерфейса 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

¹ Автор признателен профессору Е.Н. Крючковой за ценное обсуждение содержания статьи.

² Варвара Андреевна Крайванова, аспирант (krayvanova@yandex.ru).

день сфера применения компьютеров настолько широка и разнообразна, что использовать для всех задач универсальный интерфейс человеко-машинного взаимодействия неэффективно. Каждый 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.

На сегодняшний день существует три основных подхода к формализации естественного языка. Наиболее распространенный на западе подход основывается на формальных грамматиках [13, 12] и позволяет достаточно эффективно формализовать синтаксис языка, однако извлечение смысла в данной парадигме представляется проблематичным. Второй подход основан на использовании не только синтаксической информации, но и семантических словарей[5], однако составлять такие словари очень трудоемко, а чрезмерная многозначность естественного языка приводит на вычислительном уровне к комбинаторному взрыву. Третий подход - использование шаблонов - наиболее прост в автоматизации и используется для извлечения семантики из достаточно строго структурированных текстов. Применение этого подхода для сколько-нибудь широкого подмножества естественного языка проблематично, так как требуется предусмотреть все возможные способы выражения той или иной мысли[5].

Таким образом, 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 [9]) до сред программирования сверхвысокого уровня.

На практике, как правило, для каждой подобной задачи строится своя достаточно сложная семантическая модель. Тем не менее, далеко не каждая задача требует глубокого семантического анализа текста, учитывающего структуру дискурса. Во многих прикладных областях понимание текстов на уровне простейшей логики было бы достаточным. Примерами таких областей служат запросы к базам данных, интеллектуальные справочные системы, электронные персональные секретари, системы управления „Умных домов“ [8]. Заметим, что во всех этих областях сейчас используются системы управления, требующие от пользователя специальных навыков (например, знание SQL, умение пользоваться сложными графическими интерфейсами), строго говоря, неадекватных содержанию предметной области.

Цель проекта - построение модели интеллектуального адаптивного ЕЯ-интерфейса, с одной стороны, достаточно разумного и быстрого для практического использования, а с другой стороны, требующий минимальных усилий для построения базы знаний. Поскольку, как уже говорилось ранее, задача синтаксического анализа в достаточной для целей проекта степени решена для широкого круга языков, в предлагаемой модели в качестве исходных данных берутся результаты синтаксического анализа. С точки зрения трех описанных выше подходов предлагаемое решение является расширением шаблонов с минимальным синтаксическим лексиконом упрощенной структуры. В качестве представления фраз естественного языка используется несколько модифицированное функциональное представление, предложенное Тузовым [10]. Статья посвящена анализу темпоральной эффективности алгоритмов логического вывода в построенной модели.

Для иллюстрации и экспериментальной реализации в качестве предметной области нами выбрана модель системы управления „Умными домами“. Модель применима к любому языку, синтаксис которого основан на подчинительных отношениях, для определенности в качестве примера будет использован русский

язык.

1. Математическая модель ЕЯ-интерфейса

Естественный язык имеет достаточно строгую грамматическую структуру, то есть содержит обширную формальную составляющую. Эта составляющая достаточно информативна. Действительно, турист с помощью нескольких стандартных фраз и разговорника может понимать незнакомый язык на бытовом уровне, не вникая в тонкости значений слов и выражений. В этом случае человек не думает на используемом языке, то есть подходит к языковым знаниям с точки зрения формальной лексико-синтаксической структуры. При этом турист в состоянии самостоятельно обучаться: встраивать в свой лексикон новые слова и узнавать новые синтаксические конструкции.

Предлагаемая математическая модель ЕЯ-интерфейса основана на этом же принципе и состоит из следующих элементов:

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

В данной работе предлагается осуществлять перевод синтаксической конструкции с естественного языка на формальный путем замены фрагментов синтаксических конструкций на их логические следствия. При этом осуществляется не только буквальный перевод команд с естественного языка на формальный, но и логический анализ фраз, не являющихся командами. Пример, иллюстрирующий этот подход, будет приведен далее.

Пусть A - алфавит лексического уровня применяемого ЕЯ. Тогда формально словом будем называть упорядоченную последовательность непустых цепочек на этом алфавите, разделенных

пробелами. Пусть конечное множество $W \subseteq A + (A+)^*$ - лексикон модели. Слова в лексиконе W могут являться как отдельные словами ЕЯ (например: ЧАЙКА, ФЛАГ, ЯБЛОКО), так и устойчивыми словосочетаниями (например: СТИРАЛЬНАЯ МАШИНА, ТЕОРИЯ ВЕРОЯТНОСТИ). Лексикон ЕЯ представляет собой гораздо более сложную структуру, чем просто множество слов. Лексикон построенной модели учитывает два аспекта лексикона ЕЯ, непосредственно определяемые через синтаксис ЕЯ: группировка слов ЕЯ в понятия и синонимичность.

Из множества W 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 $w_B \in W$.

Мы исходим из посылки, что знания модели о предметной области не полны, поэтому небольшое начальное (созданное экспертом по знаниям) множество слов W и отношения обобщения и синонимичности уточняются в процессе функционирования модели на основе анализа позиции в фразе, где встретилось незнакомое слово. Разработанные механизмы адаптации модели в процессе функционирования описаны в [3].

Для дальнейшей иллюстрации будем рассматривать приведенный ниже начальный лексикон.

$W = \{\text{НАПИСАТЬ, НАБРАТЬ, НАПЕЧАТАТЬ, "ТЕКСТОВЫЙ ДОКУМЕНТ СТАТЬЯ, ПИСЬМО, ЗАПУСТИТЬ, БЛОКНОТ, ПРОИГРЫВАТЕЛЬ, ПРОГРАММА, ВКЛЮЧИТЬ, ПРИБОР, ПОМЕЩЕНИЕ, ТЕМНО, СВЕТИЛЬНИК, ЧАЙНИК, КОМНАТА, КУХНЯ}\}$

- $Gen(\text{СТАТЬЯ, "ТЕКСТОВЫЙ ДОКУМЕНТ"}) = 1;$
- $Gen(\text{ПИСЬМО, "ТЕКСТОВЫЙ ДОКУМЕНТ"}) = 1;$
- $Gen(\text{БЛОКНОТ, ПРОГРАММА}) = 1;$

- $\text{Gen}(\text{БЛОКНОТ}, \text{ПРОИГРЫВАТЕЛЬ}) = 1;$
- $\text{Gen}(\text{СВЕТИЛЬНИК}, \text{ПРИБОР}) = 1;$
- $\text{Gen}(\text{ЧАЙНИК}, \text{ПРИБОР}) = 1;$
- $\text{Gen}(\text{КОМНАТА}, \text{ПОМЕЩЕНИЕ}) = 1;$
- $\text{Gen}(\text{КУХНЯ}, \text{КОМНАТА}) = 1;$
- $\text{Syn}(\text{НАПЕЧАТАТЬ}, \text{НАБРАТЬ}) = 1;$
- $\text{Syn}(\text{НАБРАТЬ}, \text{НАПЕЧАТАТЬ}) = 1;$

Все прочие отношения равны нулю. Все новые для модели слова также добавляются в лексикон.

Рассмотрим представление фраз ЕЯ. В фразах естественного языка можно выделить три основных вида синтаксических отношений:

- подчинительные отношения соединяют некоторое главное слово с зависимой синтаксической конструкцией;
- сочинительные отношения - это отношения между конструкциями, синтаксически не подчиненными друг другу и связанными между собой союзом и/или интонацией (например, однородные члены предложения) [6];
- отношение отрицания (унарное) - это признак того, что утверждение не является истинным.

Для ЕЯ это отношение гораздо сложнее традиционного логического отрицания. В общем случае о значении фразы с признаком отрицания ничего конкретного сказать нельзя. Поэтому в дальнейшем мы не будем строить никаких предположений о семантике отрицания.

В предлагаемой модели за основу представления ЕЯ-фразы возьмем подчинительные синтаксические отношения. С точки зрения подчинительных отношений ЕЯ-фраза представляется

некоторой иерархической структурой, в которой главные слова находятся над зависимыми.

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. Фактически, это типы возможных в языке подчинительных отношений. На практике тип аргумента может представлять собой просто некоторую морфологическую информацию.

Чтобы абстрагироваться от порядка слов в предложении, будем считать, что на множестве Rel всех типов аргументов и на множестве всех слов W задана геделевская нумерация $Ng(r), r \in Rel$ и $Ng(w), w \in W$. Аргументы некоторого функционального символа упорядочены сначала по геделевскому номеру типа аргумента, а потом - по геделевскому номеру функционального символа - слова.

Функциональную форму фразы ϕ удобно изображать в виде дерева, вершины которого помечены словами $w \in W$, а ребра - типами аргументов $r \in Rel$ (Рис. 1). Корень дерева фразы ϕ 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 [7]. Для иллюстрации используются деревья, полученные этим анализатором. К предлагаемому представлению с помощью достаточно очевидных преобразований приводятся такие традиционные

представления синтаксических конструкций ЕЯ, как граф зависимостей или дерево непосредственных составляющих [5].

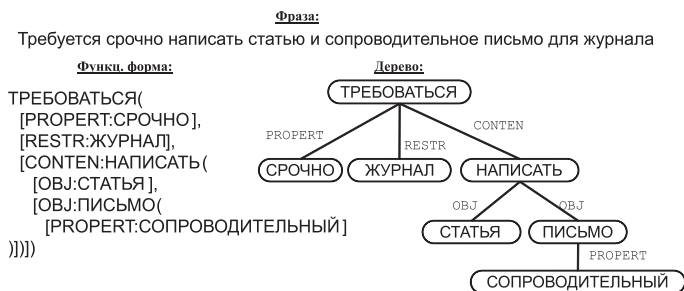


Рис. 1. Пример построения дерева фразы

Команда системы управления представляется парой:

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

где ν - функциональная форма, задающая определение команды на ЕЯ. Некоторые функциональные символы в ν 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 общем случае команда - это некоторая фиксированная параметризованная синтаксическая конструкция, понятная объекту управления. Можно показать, что общность модели не нарушается, вне зависимости от того, используется ли в командах формальный или естественный язык. Для определенности будем считать, что команда в модели - это фиксированное описание формальной команды объекта управления на естественном языке, то есть используются типы аргументов и слова естественного языка,

В узлы дерева ν , помеченные параметром $name_i$, может быть подставлено некоторое дерево ϕ , 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 ϕ 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 γ , который определяет степень уверенности системы в наличии команды.
- множество $Values = \{ \langle p_i, \phi_i \rangle \}$, где p_i - параметры команды e , ϕ_i - значения параметров. Причем для каждого параметра p_i должно существовать единственное значение ϕ_i .

Обозначим $Value(p_i) = \phi_i$.

Задача логического анализа заключается в том, чтобы свести некоторую начальную фразу ϕ 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 ν и σ - 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 ν - 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 ν - функциональная форма, представляющая шаблон команды объекта управления или фразу-ответ на ЕЯ.

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

На основе введенных определений задачу логического анализа фразы ϕ можно сформулировать следующим образом. **Задача 1.** Найти множество E_ϕ всех заключительных функциональных форм, выводимых из данной функциональной формы ϕ с помощью цепочек вывода длины не более $Lenght$. Очевидно, что при такой постановке задачи множество E_ϕ конечно.

Далее приведен пример логического анализа нескольких фраз. Объект управления в примере умеет только включать известные ему приборы (светильник и чайник), расположенные в известном помещении (на кухне) и запускать известные программы (блокнот и проигрыватель). Для краткости будем опускать обозначение типа аргумента.

$$Rules = \{$$

- 1) ЗАПУСТИТЬ (\$X) $Parameters = \{\{ \$X, "ПРОГРАММА" \} \};$
- 2) ВКЛЮЧИТЬ (\$X, \$Y) $Parameters = \{\{ \$X, "ПОМЕЩЕНИЕ" \}, \{ \$Y, "ПРИБОР" \} \};$
- 3) НАПИСАТЬ (\$X) $\rightarrow$ НАБРАТЬ (\$X), $Parameters = \{\{ \$X, "ТЕКСТОВЫЙ ДОКУМЕНТ" \} \};$
- 4) НАБРАТЬ (\$X) $\rightarrow$ ЗАПУСТИТЬ (БЛОКНОТ), $Parameters = \{\{ \$X, "ТЕКСТОВЫЙ ДОКУМЕНТ" \} \};$
- 5) ТЕМНО (\$X) $\rightarrow$ ВКЛЮЧИТЬ (\$X, СВЕТИЛЬНИК) $Parameters = \{\{ \$X, "ПОМЕЩЕНИЕ" \} \};$

}

Рассмотрим анализ фразы "Нужно срочно написать статью для журнала". Заметим, что эта фраза не является явной командой объекту управления. Функциональная форма, представляющая фразу: НУЖНО(СРОЧНО, ЖУРНАЛ, НАПИСАТЬ(СТАТЬЯ)). Применение правила 3 к функциональной форме: НУЖНО(СРОЧНО, ЖУРНАЛ, НАБРАТЬ(СТАТЬЯ)). Применение правила 4: НУЖНО(СРОЧНО, ЖУРНАЛ, ЗАПУСТИТЬ(БЛОКНОТ)). Применение заключительного правила 1 дает команду: ЗАПУСТИТЬ(БЛОКНОТ). Аналогичным образом фраза "В кухне слишком темно" приведет к команде ВКЛЮЧИТЬ(КУХНЯ, СВЕТИЛЬНИК). Таким образом, предложенная модель позволяет описывать принципиально более сложную логику, чем простая интерпретация команд, вроде "Включи чайник". Пример показывает, что адекватность результатов логического вывода определяется наполнением множества правил и отношениями на лексиконе, и не зависит от используемого языка.

Однако, алгоритмы, решающие Задачу 1, имеют высокую временную сложность.

Теорема: Задача 1 является NP-трудной.

Доказательство теоремы можно найти в [2].

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.

2. Проблема увеличения скорости работы алгоритма логического анализа

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 ϕ 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.

3. 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.

В качестве анализируемого языка использован русский. Анализируемые предложения имеют различную сложность и длину от 2 до 11 слов. Заметим, что фразы на естественном языке редко бывают длиннее 20 слов, поэтому их размер мало влияет на общую производительность алгоритмов модели.

Знания модели в проводимых экспериментах являются полными, то есть при унификации уровень релевантности результата равен 0 или 1, и обучение системы не производится.

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 ветвящихся 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.

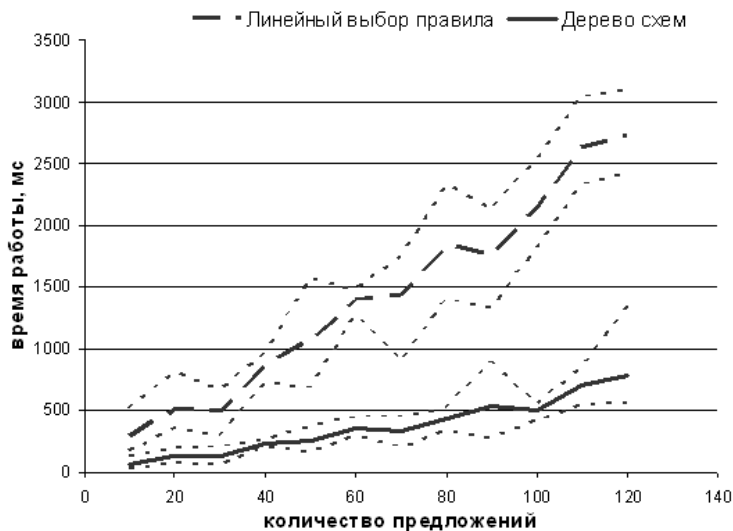


Рис. 2. Сравнение алгоритмов выбора правила ЛВП и ДС.
Зависимость от размера текста.

- Максимальная глубина дерева вывода - 5 шагов.
- Размер текста: 128 предложений.

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.

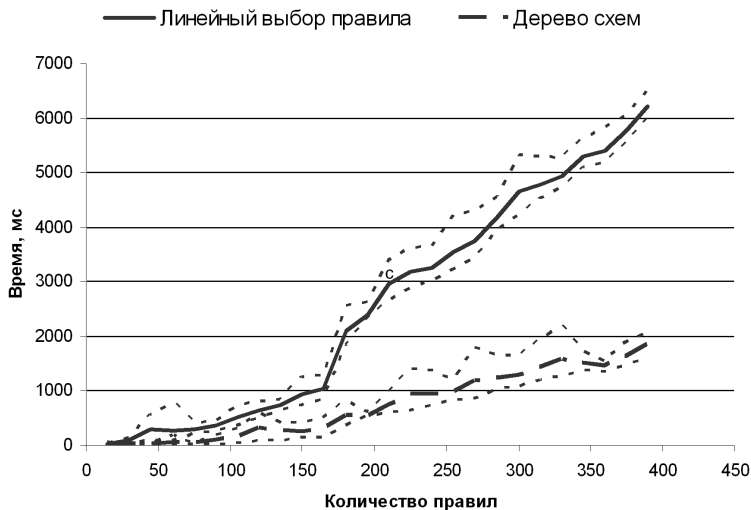


Рис. 3. Сравнение алгоритмов выбора правила ЛВП и ДС.
Зависимость от количества правил.

- Диапазон изменения размера текста: от 10 до 120 предложений; шаг: 10 предложений.

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.

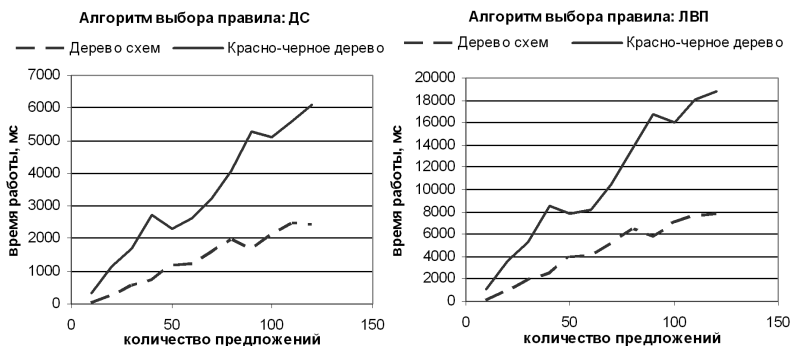


Рис. 4. Сравнение ДС и красно-черного дерева как структур данных для хранения полученных результатов (для предотвращения повторного вывода) для различных способов выбора правила: ЛВП и ДС.

- 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.

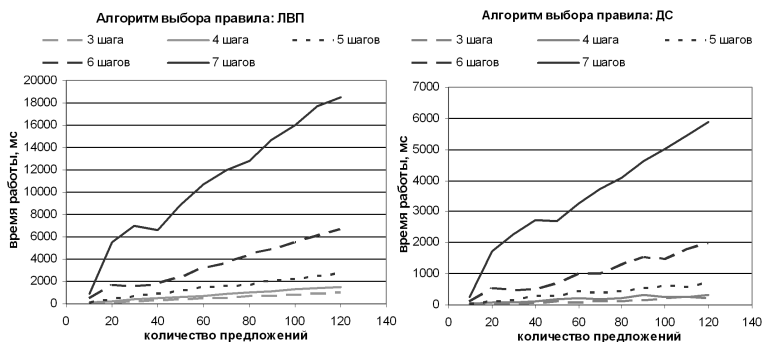


Рис. 5. Зависимость времени работы алгоритмов от глубины вывода для различных способов выбора правила: ЛВП и ДС.

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

4. Общая архитектура системы

В качестве языка программирования для реализации предложенной математической модели выбран мультиплатформенный язык Java. Взаимодействие модулей реализации системы представлено на рисунке 6[4].

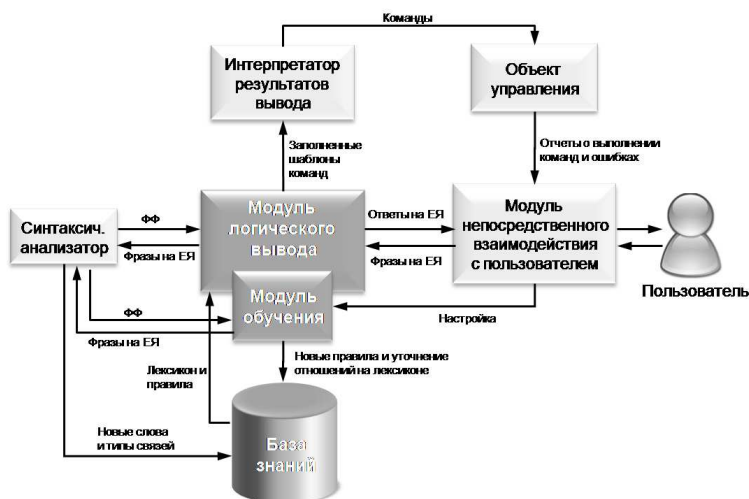


Рис. 6. Схема потоков данных системы.

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 ко-

манд 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 обучения автоматизирует процесс построения базы знаний системы.

5. Выводы

Вычислительные эксперименты демонстрируют, что предложенная математическая модель адекватна моделируемому объекту в пределах поставленной задачи. Тестирование производительности показало высокую эффективность механизмов повышения темпоральной эффективности. 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. ЛЮГЕР Д. Ф. *Искусственный интеллект: стратегии и методы решения сложных проблем*. 4-е издание. - М.: Вильямс, 2003.
2. КРАЙВАНОВА В. А. *Вычислительная сложность задачи логического анализа для адаптивно-логической модели естественного языка*. Материалы двенадцатой региональной конференции по математике "МАК-2009". - Барнаул: издательство АГУ, 2009. - стр. 130-132.
3. КРАЙВАНОВА В. А., КРЮЧКОВА Е.Н. *Концепция адаптивного управления интеллектуальными объектами на основе логического вывода*. Ползуновский альманах №4/2008. - Барнаул: издательство АлтГТУ. - стр. 26-30.
4. 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.
5. НАЙХАНОВА Л.В., ЕВДОКИМОВА И.С. *Методы и алгоритмы трансляции естественно-языковых запросов к базе данных в SQL-запросы*. Улан-Уде: Издательство ВСГТУ, 2004.
6. РОЗЕНТАЛЬ Д.Э. *Словарь лингвистических терминов*.
7. СОКИРКО А. В. *Реализация первичного семантического анализа в системе Диалинг*. Тр. междунар. семинара Диалог'2000 по компьютерной лингвистике и её приложениям, - Протвино, 2000.
8. *Сайт компании House Control*. Режим доступа: <http://www.housecontrol.ru>
9. *Сайт Российского НИИ искусственного интеллекта*. Режим доступа: <http://artint.ru>
10. ТУЗОВ В.А. *Компьютерная семантика русского языка*

СПб.: Изд-во СПбГУ, 2004. – 400 с.

11. ХОРОШЕВСКИЙ В.Ф. *Оценка систем извлечения информации из текстов на естественном языке: кто виноват, что делать* // Труды Десятой национальной конференции по искусственному интеллекту с международным участием (КИИ- 2006). - М.: Физматлит. - 2006. - Т. 2. - С. 464-478.
12. ЧЕПОВСКИЙ А.М. *Неразрешимая проблема компьютерной лингвистики* // "Компьютерра" №30 от 02 августа 2002 года.
13. CHOMSKY N. *The logical basis of linguistic theory* // Preprints of papers for ninth international congress of linguistics. - Cambridge: Mass. - 1962.

THE MODEL OF NATURAL LANGUAGE INTERFACE FOR CONTROL SYSTEMS OF COMPLEX TECHNICAL OBJECTS AND EFFICIENCY EVALUATION OF MODEL-BASED ALGORITHMS

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

Abstract: The article consider the problem of construction of intelligent computer human interface based on natural language phrases logic analysis. The mathematical model of interface 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.