Министерство образования и науки Российской Федерации
Московский государственный технический университет им.Н.Э.Баумана
Кафедра «Системы обработки информации и управления»

ИНТЕЛЛЕКТУАЛЬНЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

Сборник учебно-методических работ и статей аспирантов и студентов

Выпуск 9

Под редакцией Ю.Н.Филипповича

НОК «CLAIM»
Москва
2007

УДК 007.52
ББК 32.813
И 76

Составитель: Ю.Н.Филиппович
Авторы статей:

А.А.Александрова, Ван Чэнь, Е.А.Выломова, Гуань Ли, И.В.Даньшина, М.В.Даньшина, С.А. Иванов, К.Е.Иовков, С.А.Кесель, Т.В.Колобкова, А.А.Кононков, Ле ба Хонг Минь, А.И.Панченко, Д.В.Пономарев, А.А. Проскурнин, А.В.Сиренко, А.Ю.Филиппович, Ю.Н.Филиппович.

И 76 **Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. Выпуск 9** / Сост. и ред. Ю.Н.Филипповича. — М.: НОК «CLAIM», 2007. — 303 с.

Сборник содержит 16 научных статей аспирантов и студентов. Статьи посвящены: исследованию и разработке информационных технологий в области образования, компьютерной лексикографии и семиотике, проектированию баз знаний и компьютерной семиографии.

Сборник предназначен для преподавателей, аспирантов и студентов специальностей двух направлений: 230100 — Информатика и вычислительная техника, 230200 — Информационные системы.

Содержание

ПРЕДИСЛОВИЕ

РЕДАКТОРСКАЯ СТАТЬЯ

СТАТЬИ АСПИРАНТОВ И СТУДЕНТОВ

АВТОРЕФЕРАТЫ КВАЛИФИКАЦИОННЫХ РАБОТ

МЕТОДИЧЕСКИЕ РАБОТЫ

Предисловие

В девятом выпуске сборника представлены научные статьи преподавателей, аспирантов и студентов кафедры «Системы обработки информации и управления (ИУ5) Московского государственного технического университета им. Н.Э.Баумана.

Редакторская статья посвящена рассмотрению вопросов моделирования вербального сознания носителей русского языка культуры на основе двух экспериментов — ассоциативного и когнитивного. Ее следует рассматривать как продолжение редакционной статьи выпуска ИТС 8.

Начало исследованиям, на основе которых написана статья, было положено несколько лет тому назад и связано с научными идеями чл.корр. РАН, д.ф.н., профессора Ю.Н.Караулова.

В статье рассмотрены два режима работы когнайзера — модельного представления функционирования вербального сознания — пассивный и активный. Предложено графовое описание модели.

Статьи аспирантов и студентов посвящены: исследованию и разработке информационных технологий и конкретных систем.

Две статьи сборника непосредственно связаны с темой когнитивных исследований и разработок. При проведении исследований, представленных в редакторской статье, использовалось инструментальное программное средство, которое описывается в статье А.И.Панченко. Статья А.А.Проскурнина посвящена проблеме извлечения теоретических знаний субъекта, относящихся к определенной предметной области в процессе взаимодействия субъект-компьютер.

В статьях китайских студентов Ван Ченя и Гуань Ли представлены разработки китайско-русского электронного словаря метафор информационных технологий. Вьетнамский студент Ле Ба Хонг Минь в своей статье описывает состав и структуру системы автоматического перевода с вьетнамского на русский язык, проектируемую им в рамках магистерской диссертации. создания А.В.Сиренко подробно описывает рабочий вариант базы данных лингвокультурного тезауруса русского языка.

Сборник содержит четыре статьи студентов, представляющие работы, выполненные под руководством доцента А.Ю.Филипповича. Две из них посвящены компьютерной семиографии — фактически складывающемуся новому направлению обработки древнерусского певческого и музыкального наследия. Авторы этих статей: Е.А.Выломова, И.А.Даньшина и М.А.Даньшина. В статье С.А.Иванова рассмотрен новый для существующей отечественной системы высшего образования и

актуальный в свете ведущихся работ по переходу на двух-уровневую систему подготовки кадров в вузах компетентностный подход к созданию систем управления знаниями в области ИКТ. К.Е.Иовков в своей статье подробно описывает возможность использования средств СУБД при проектировании и реализации компонент экспертной системы

В статьях Т.В.Колобковой, М.А.Павловой Д.В.Пономарева, А.С.Сигачева рассматриваются различные вопросы обработки и представления в электронном виде текстовой информации, хранящейся в библиотеках и архивах.

В специальный раздел сборника собраны квалификационные работы студентов: авторефераты дипломных проектов А.А.Александровой и А.В.Сиренко и магистерской диссертации А.А.Кононкова.

В методическом разделе сборника напечатаны разработки доцентов А.Ю.Филипповича и С.А.Кеселя по дисциплинам «Архитектура АСОИУ» и «Защита информации в АСОИУ», которые читаются студентам кафедры ИУ-5.

Благодарю за помощь в издании сборника Г.А.Черкасову и А.Л.Волкова.

Ю.Н.Филиппович

Редакторская статья

Ю.Н.Филиппович

МОДЕЛИРОВАНИЕ ЯЗЫКОВОГО СОЗНАНИЯ[1]

Введение

Языковое сознание (ЯС) – это форма мышления человека с использованием языковых единиц (ЯЕ). ЯС возникает и развивается как процесс *осознавания* — перехода от неосознанного (неопределенного, «альтернативного») восприятия предмета реального мира к осознанному (определенному, «безальтернативному»). Предмет осознан — значит вербализован, ему в соответствие поставлена некоторая конкретная языковая единица. ЯС локализовано в субстрате мышления и неотрывно от него. Реально существующее в пространственно-временном континууме оно может быть измерено. Пространственными свойствами размерности ЯС являются его элементарность (дискретность и непрерывность), структурность и протяженность; временными — длительность, неповторимость и необратимость. ЯС проявляет себя симультанно в виде пространственного *объекта* знаний о мире (языковой картины мира — ЯКМ); в виде развивающегося во времени *процесса* порождения единиц знаний о мире (ЕЗМ); в виде *ситуации=явления* (композиции объекта и процесса) — как некоторый когнайзер, динамично изменяющий симультанную ЯКМ посредством двух разнонаправленных процессов: от языковой единицы к знаниям о мире (ЯЕ → ЗМ) и от знаний о мире к языковой единице (ЗМ → ЯЕ).[2]

[1] В статье представлены результаты работ, выполняемых по проекту РФФИ 05-06-80284 «Языковое сознание нашего современника: когнитивная структура и лингвокультурное содержание».

[2] Эти суждения представляют собой попытку непротиворечивого соединения авторских представлений о предмете и исходных посылок-определений Ю.Н.Караулова для тех основных понятий, которые используются в данной статье: «…*языковое сознание* складывается из вербально выраженных знаний о мире, т.е. содержанием языкового сознания является вербализованная часть картины мира. ... *языковое сознание представляет собой подвижное, динамическое образование, своего рода когнайзер, манипулирующий элемен-*

Правомерность таких суждений о ЯС требует экспериментального подтверждения, которое состоит из собственно экспериментов, сводящихся к наблюдению, измерению и фиксации его проявлений, и последующих модельных построений.

Экспериментов два.

Первый — это свободный вербальный ассоциативный эксперимент, результатом которого является ассоциативный тезаурус — АТ (ассоциативная вербальная сеть — АВС)[3], моделирующий активный режим работы ЯС, и построенный на основе пар слов-стимулов и слов-реакций <S→R>. По условиям эксперимента исходным является предъявляемое испытуемому слово (стимул), т.е. ЯЕ, и предлагается «ответить» на это слово любым спонтанно и первым пришедшим в голову другим словом или словосочетанием. Результатом эксперимента становится соединение двух языковых единиц, которое приобретает новое качество: полученная пара стимул-реакция несет знания о мире, превращается в элементарную единицу знаний о мире (ЕЗМ).

Второй — это когнитивный эксперимент (языковая игра типа «кроссворд»), результатом которого является когнитивный тезаурус и построенный на основе когнем. По условиям этого эксперимента испытуемому предъявляется естественно-языковая конструкция (вербальная формула смысла — ВФС), состоящая из нескольких ЯЕ, и предлагается «ответить» на эту конструкцию языковой единицей — словом-знаком (Зн), отражающим ее смысл. Результатом эксперимента становится более сложная конструкция, состоящая из ВФС и Зн, также несущая знание о мире, и являющаяся его единицей (ЕЗМ).

Модельными сущностями являются: а) языковые картины мира, б) процессы вербализации знаний о реальном мире, в) когнайзер.

Языковые картины мира представляются в форме баз данных ассоциативного и когнитивного экспериментов.

тарными единицами знания (фигурами знания) и функционирующий в активном, смыслопорождающем (т.е. в направлении от знака — к смыслу) и пассивном, знакопорождающем (от смысла — к знаку) режимах. … Языковая, или наивноязыковая, картина мира складывается не из слов, не из понятий, имеющих логико-лингвистическую природу, а из единиц когнитивной природы, обладающих различной системообразующей мощностью и вступающих одна с другой в различные иерархически-координативные отношения …» [Караулов 2004 а]. Понятие элементарной единицы знаний — «фигуры знания», «когнемы» — введено и интерпретировано как минимальная когнитивная единица, представляющая собой пятикомпонентное отношение {<слово-знак>, <вербальная-формула-смысла>, <способ задания смысла>, <референтная область>, <функция>} или пентаграмму — полносвязанный пятивершинный граф. Подробно см. работы [Караулов 2003а,б,в].).

[3] См. например работы Ю.Н.Караулова, Ю.А.Сорокина, Е.Ф.Тарасова, Н.В.Уфимцевой, Г.А.Черкасовой по теме Русский ассоциативный словарь.

Процессов вербализации знаний о реальном мире два: первый из них (от слова к знанию) квалифицируется как активный режим работы когнайзера — он ориентирован на познание и состоит в развертывании некоторой языковой единицы в единицу знаний о мире; а второй (от знания к слову) — как пассивный, ориентированный на мышление[4] и состоящий в свертывании единицы знаний о мире в некоторую языковую единицу.

Когнайзер — база знаний, интегрирующая процессы вербализации знаний и базы данных ассоциативного и когнитивного экспериментов в форме процедуры осознавания — принятия решения о выборе вербальных альтернатив представления знаний о реальном мире.

Выводы, которые были сделаны из этих экспериментальных наблюдений и построений таковы [Караулов 2004 б]:

- осознавание является той процедурой, которая преобразует семантические отношения в когнитивные, превращая единицы языка (слово, словосочетание, предложение) в простейшие единицы знания;

- такое преобразование осуществляется в разной форме в зависимости от режима, в каком работает языковое сознание: в активном режиме осознавание происходит путем построения ассоциативной цепочки из стимулов и реакций в АВС, цепочки, из которой формируется пропозиция; в пассивном режиме тот же процесс реализуется в фигуре знания;

- знание, которое извлекается из АВС чаще бывает не очень точным, несколько размытым и неоправданно детализированным;

- тем не менее, между двумя типами знания практически всегда устанавливаются отношения эквивалентности, если только подвергающийся осознаванию знак присутствует в числе стимулов или реакций в АВС.

Взяв за основу сделанные выводы, попробуем сконструировать формальную модель процедуры осознавания в когнайзере для последующей реализации в виде компьютерной программы.

Формальное моделирование

Первоначально, преследуя только методические цели, уточним (упростим) содержание введенного ранее понятия «осознавание». Сведем процесс осознавания (переход от неосознанного к осознанному = от

[4] Использование категорий «познание» и «мышление» в данном контексте обусловлено их введением в раннее опубликованных статьях. Они являются составными частями «универсального категориального аппарата», связывающего два диалектических единства <часть–целое> и <форма–содержание>.

невербализованного к вербализованному) к безальтернативным и альтернативным вариантам.

Безальтернативные варианты осознавания — это переходы от априори известных ЕЗМ к ЯЕ и обратно.

Альтернативные варианты характеризуются вариативностью, неопределенностью (вероятностью) и неточностью этих переходов. Иначе, имеется множество ЕЗМ и ЯЕ, а также процедура перехода между ними, которая состоит в выборе альтернативных единиц, традиционно разделяемая на две составные части — критериальная оценка альтернатив и принятие решения о предпочтении (выбор альтернатив).

Далее наибольшее внимание уделим формальному описанию безальтернативных вариантов осознавания, и только наметим возможные подходы к моделированию альтернативного осознавания.

Рассмотрим самую простую и интуитивно понятную модель: представим ассоциативно-вербальную сеть в виде графа, вершинами которого являются слова (в общем случае некоторые языковые единицы — ЯЕ), а дугами выявленные в эксперименте отношения между ними. Все множество вершин графа можно разделить на три типа: слова-стимулы (S), слова-реакции (R), слова-стимулы-реакции (SR). Отношения между ними устанавливаются экспериментально, если респондент одному слову, называемому словом-стимулом ставит в соответствие другое слово, которое впоследствии рассматривается как слово-реакция. Данное отношение будем называть ассоциативным и обозначать символами «→», «←», «↔».

Результатом эксперимента являются следующие подмножества пар слов: {S, R}, {S, SR}, {SR, R}, {SR, SR}. Заметим, что в процессе эксперимента конкретное отношение может быть установлено разнонаправлено, т.е. одно и тоже слово может оказаться и словом-стимулом и словом реакцией. Кроме этого одно и тоже отношение может быть установлено несколько раз, в этом случае будем называть частоту встречаемости отношения его валентностью и указывать ее численное значение. В первых трех из указанных подмножеств следующие отношения <S→R>, <S→SR>, <SR→R>, <S←R>, <S←SR>, <SR←R> только одновалентны; в последнем подмножестве отношения могут быть одновалентны <SR→SR> и бивалентны <SR↔SR>. Это означает, что в графовой модели АВС есть следующие типы вершин: а) вершины S (корневые вершины, корни), имеющие только выходящие дуги, связывающие их с вершинами типа R и SR; б) вершины типа R (листьевые вершины, листья), имеющие только входящие дуги, связывающие их с вершинами

типа S и SR; в) вершины типа SR, имеющие как входящие, так и выходящие дуги, связывающие их с вершинами типа S, R и SR.

В Русском ассоциативном словаре зафиксировано 103211 слов (языковых единиц), в том числе: слов-стимулов {S}— 160; слов-реакций {R}— 96587; остальных слов, которые являются и словами-стимулами и словами-реакциями {SR} — 6464.

В графовой модели АВС для двух любых вершин можно установить связи, которые будут представлять собой пути в графе, или цепочки, состоящие из последовательности вершин и дуг. Анализируя возможные цепочки, выделим два типа: *однонаправленные* — вершины цепочки связаны между собой только одновалентными отношениями <S→R>, <S→SR>, <SR→R>, <SR→SR>; *разнонаправленные* — вершины связаны всеми возможными одновалентными и бивалентными отношениями.

Особый тип цепочек представляют собой кольца. В цепочках, как первого, так и второго типов могут присутствовать «закольцованные участки» различной длины, под которой будем понимать количество однонаправленных отношений, приводящих к исходной вершине. Единичное кольцо — это замыкание вершины на саму себя. Ниже будут приведены несколько примеров цепочек с закольцованными участками.

Нахождение в АВС цепочек первого и второго типа представляет собой моделирование процедуры осознавания в когнайзере, т.е. его функционирование в активном и пассивном режимах. Построим простые графовые модели пропозиций формул смысла нескольких когнем, т.е. смоделируем в АВС конкретные варианты пассивного и активного режимов работы когнайзера.

Пассивный режим работы когнайзера.

Начнем с нахождения однонаправленных цепочек. В качестве примера возьмем когнему «Арбалет»:

Знак = *Арбалет.* Формула смысла = *Старинное оружие в форме лука*.

Первоначально определим пропозицию формулы смысла как:

<старинное> | *<оружие>* | *<в форме>* | *<лука>*.

В АВС элементам пропозиции *старинное*, *лука* соответствуют только листьевые вершины графа типа R, а конструкция *в форме* отсутствует, т.е. для данной формулы смысла мы имеем только одну возможную исходную вершину — *оружие*. Эта вершина относится к типу SR, т.к. имеет более восьмидесяти входных дуг, являясь реакцией соответствующего количества слов-стимулов, и более сотни выходных, порождая слова-реакции (см. словарные статьи обратного и прямого РАС).

Оружие (обр.)

ОРУЖИЕ* огнестрельное **93;** применять **10;** сдать **9;** пистолет, пушка **7;** пулемет **6;** заряжать, стрельба **5;** булыжник, кинжал, носить, сдавать, ствол **4;** Калашников, слово, шпага **3;** автомат, бросать, войска, вооружен, древнее, копье, ликвидировать, орудийный, патрон, ружье, убийство **2;** армия, атаковать, атом, атомная бомба, атомный, байки, бандит, битва, боец, болванка, бомба, борьба, бумага, везти, вершина, военный, воин, Вторая мировая война, выбросить, выстрел, град, дубина, защитник, инструмент, клинок, конструктор, личный, лук, наше, нужно, оборона, отнять, отобрать, птица, ракета, ржавое, самоубийство, склад, смертельный, создавать, солдат, спрятать, танк, убивать, убийца, уничтожить, хранение, хранить, цели, цепь, чистить, юмор, ядерный, ядро **1;** ***81+247***

Оружие (пр.)

ОРУЖИЕ: холодное **9;** массового поражения, *ружье* **5;** *огнестрельное,* ядерное **4;** *война,* стреляет, убийства **3;** *безопасность,* мощное, *пистолет,* смертельное, *смерть,* старинное, убийцы **2;** абсолютное, *автомат, армия,* Бальзак, винтовка, военное, возмездия, врага, *в руках,* выстрелило, газовое, грозное, *дерево, железо, зонтик,* именное, и пушка, *кинжал,* командира, *кровь,* любви, массового уничтожения, *мести, мое,* мортира, МП, на складе, *нож, опасно,* опасное, *орудие,* перестройки, *перо,* пищаль, *продавать,* пролетариата, *прощай, прятать,* разоружение, самозащиты, секретное, *сильный, слово,* сложить, смерти, спортивное, *ствол, стрельба, стрелять,* твое, холодная, *черный* **1;** ***105+67+3+52***

Нас будут интересовать только вершины, являющиеся словами-реакциями. Найдем среди них такие, которые означают слова *стрельба и орудие*. Дуги, связывающие их, имеют валентность равную 1. Эти вершины также относятся к типу SR. У них 21 и 15 входных и 103 и 106 выходных дуг соответственно. Среди них есть такие, которые связывают их с вершиной *арбалет,* валентности связей равны 1.

Стрельба (пр.)

СТРЕЛЬБА: из лука **24;** *пистолет* **7;** *оружие* **5;** из пистолета, *лук, мишень,* по мишени, *ружье* **4;** *война* **3;** *автомат,* из автомата, меткая, *убийство* **2;** автоматная, арбалет, винтовка, влет, вслепую, в тире, в цель, выстрелы, гул, духарик, жертва, *заяц,* идиотство, из винтовки, из орудия, из оружия, из пулеметов, индейцы, *кровь,* кутила, меткость, наугад, огнеметная, *огонь,* пальба, перекрестная, *повсюду,* по живым мишеням, по мишеням, *пулемет,* пули, *пуля,* револьвер, рельба, сильная, стрелец **1;** ***103+49+0+36***

Орудие (пр.)

ОРУДИЕ: *труда* **40;** убийства **17;** *убийство* **8;** *пушка, труд* **4;** *кинжал, лопата* **2;** арбалет, артиллерист, боя, войны, *выстрел,* для защиты, *древнее, дым,* железное, захвата, к бою, массового поражения, метательное, *мир, молоток, нож, огнестрельное, пистолет, плуг, предмет,* предмет деятельности, преступления, производства, пролетариата, пыток, *работа, сигарета,* смерти, *топор* **1;** ***106+36+0+29***

Вершина *арбалет* относится к типу R, и является в рассматриваемой когнеме знаком. Таким образом, мы установили две цепочки, связывающие выделенный элемент пропозиции со знаком. Длина этих цепочек равна двум отношениям:

<*оружие* $\rightarrow^{1}$ *стрельба* $\rightarrow^{1}$ *арбалет*>;

<*оружие* →[1] *орудие* →[1] *арбалет*>.

Заметим, что в рассматриваемом примере вершина *арбалет* связана только с двумя вершинами — *стрельба* и *орудие* (см. обратный РАС):

Арбалет (обр.)

АРБАЛЕТ орудие, стрельба **1; *2+2***

Среди слов-реакций, порожденных словом-стимулом *оружие,* есть, например, *пистолет*, *ружьё*, *ствол*. Для первых двух словом-реакцией является *стрельба*, а для последнего — *орудие*. Так мы получаем еще три цепочки, но длиной в три отношения:

<*оружие* →[2] *пистолет* →[2] *стрельба* →[1] *арбалет*>;

<*оружие* →[5] *ружьё* →[1] *стрельба* →[1] *арбалет*>;

<*оружие* →[1] *ствол* →[1] *орудие* →[1] *арбалет*>.

Действуя таким образом, найдем все цепочки длиной максимум в 4 отношения от элемента пропозиции *оружие* до вершин *стрельба* и *орудие*, это эквивалентно нахождению цепочек длиной в 5 отношений до вершины *арбалет*:

Длина	Путь
	оружие
1	оружие —1—> стрельба
2	оружие —2—> пистолет —2—> стрельба
2	оружие —5—> ружье —1—> стрельба
3	оружие —5—> ружье —5—> охота —1—> стрельба
	…
3	оружие —1—> армия —1—> оружие —1—> стрельба
4	оружие —1—> армия —1—> мучение —1—> борьба —1—> стрельба
	…
4	оружие —1—> ствол —1—> пушка —1—> чугун —1—> стрельба
	Всего цепочек 141, в том числе: 1 — нет, 2 — 1, 3 — 2, 4 — 15, 5 — 123.

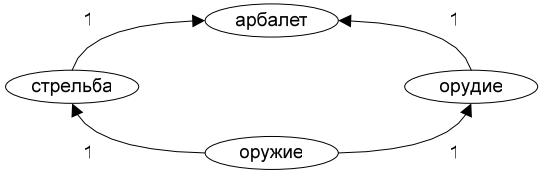
2	оружие —1—> ствол —1—> орудие
3	оружие —5—> ружье —1—> ствол —1—> орудие
3	оружие —1—> перо —1—> топор —1—> орудие
4	оружие —2—> пистолет —1—> железо —3—> лом —1—> орудие

	...
4	оружие —1—> армия —28—> солдат —1—> топор —1—> орудие
	Всего цепочек 29, в том числе: 1 — нет, 2 — нет, 3 — 1, 4 — 2, 5 — 26

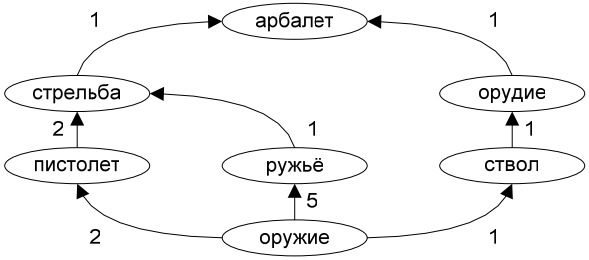
В результате нашего дальнейшего поиска непременно будут найдены и более длинные цепочки. Общее их количество будет определяться конкретными отношениями, зафиксированными в ассоциативном эксперименте.

Для заданной длины возможных цепочек, связывающих элемент пропозиции со знаком когнемы в АВС можно выделить подграф (рис. 1) соответствующей размерности, который будем рассматривать как конкретное модельное представление пассивного режима работы когнайзера. Такое представление позволяет определить для последующих рассуждений: а) цепочки и подграфы отдельного элемента пропозиции — элементарные пропозиционные цепочки и графы; б) размерности цепочек и графов, например, длины и валентности отношений, количество вершин и дуг; в) структурные особенности цепочек, например количество закольцованных участков, типы и размеры колец.

Размерность 2.



Размерность 3



Размерность 4

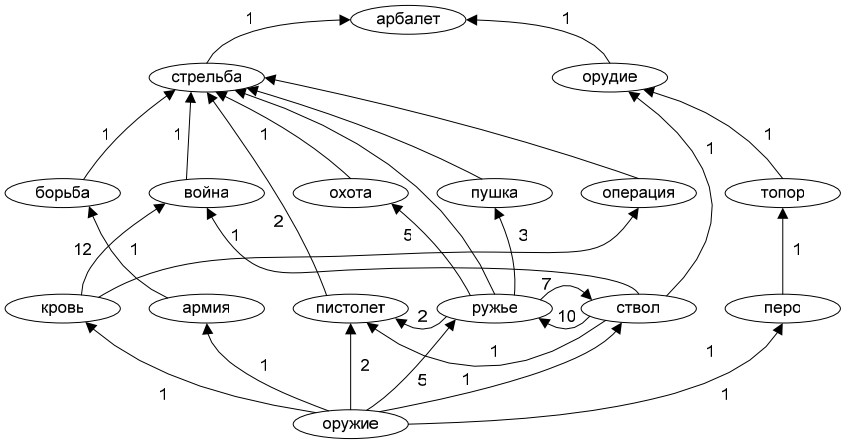


Рисунок 1. Подграфы пропозиционного элемента *оружие*.

В рассмотренной нами пропозиции, ее элементами являлись в основном словоформы. Формальное их несовпадение со словами, зафиксированными в АВС, оказало влияние на количество вершин графа, которые были выбраны в качестве начальных при поиске цепочек, а в итоге и на размерность подграфа, моделирующего когнайзер. Анализ слов АВС показывает, что в ней есть слова, которые являются основной формой для пропозиционных элементов и одновременно относятся к типу SR, например, *старинное* — ***старинный***, *форме* — ***форма***, *лука* — ***лук***. Сформируем новую пропозицию, применив к ранее выделенным элементам процедуру лемматизации — переводу слова в основную форму:

<***старинный***> | <***оружие***> | <***форма***> | <***лук***>.

Найдем в графе АВС для каждого из пропозиционных элементов цепочки длиной не более 5, связывающие их с вершиной *арбалет*. В результате получим пропозиционный подграф рассматриваемой когнемы, который будет являться моделью пассивного режима работы когнайзера (рис. 2).

Длина	Путь
	старинный
3	старинный —1—> двор —1—> ружье —1—> стрельба

4	старинный —4—> комод —1—> зеркало —1—> взгляд —1—> стрельба
	…
4	старинный —2—> образ —1—> мысль —1—> стрела —1—> стрельба
	Всего цепочек 31, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 1, 5 — 30

3	старинный —1—> двор —1—> ствол —1—> орудие
4	старинный —1—> двор —7—> вор —1—> лом —1—> орудие
	…
4	старинный —1—> двор —2—> хозяйственный —1—> топор —1—> орудие
	Всего цепочек 22, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 1, 5 — 21
	форма
2	форма —1—> спорт —1—> стрельба
2	форма —1—> война —1—> стрельба
3	форма —1—> солдат —2—> ружье —1—> стрельба
	…
3	форма —1—> армия —1—> оружие —1—> стрельба
4	форма —1—> война —1—> убийца —1—> боевик —1—> стрельба
	…
4	форма —2—> страшная —1—> мысль —1—> стрела —1—> стрельба
	Всего цепочек 103, в том числе: 1 — нет, 2 — нет, 3 — 2, 4 — 4, 5 — 98

3	форма —1—> солдат —1—> топор —1—> орудие
4	форма —1—> спорт —1—> жлоб —1—> лом —1—> орудие
	…
4	форма —1—> армия —28—> солдат —1—> топор —1—> орудие
	Всего цепочек 10, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 1, 5 — 9

	лук
2	лук —3—> стрела —1—> стрельба
4	лук —1—> стук —1—> окно —1—> взгляд —1—> стрельба
	…

4	лук —1—> салат —2—> лук —3—> стрела —1—> стрельба
	Всего цепочек 21, в том числе: 1 — нет, 2 — нет, 3 — 1, 4 — нет, 5 — 20
3	лук —1—> перо —1—> топор —1—> орудие
3	лук —1—> стук —1—> топор —1—> орудие
4	лук —1—> салат —1—> солдат —1—> топор —1—> орудие
4	лук —1—> стук —1—> молоток —10—> топор —1—> орудие
	Всего цепочек 4, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 2, 5 — 2

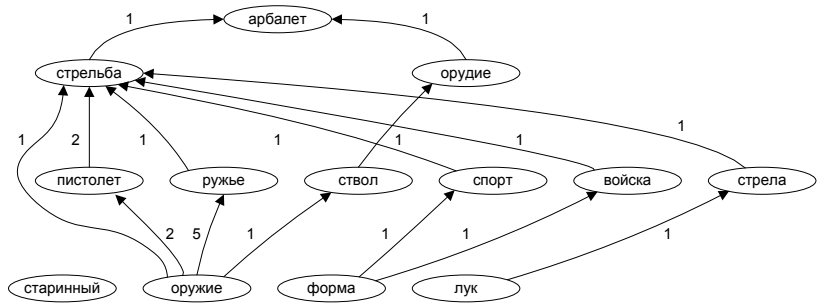


Рисунок 2. Пассивный пропозиционный граф
размерности 3 когнемы «Арбалет»

Анализ графа на рис. 2 показывает, что пропозиционный элемент *старинный* не связан цепочками длиной 3 со знаком когнемы «Арбалет». Слова *старинный* и *арбалет* в АВС связаны между собой цепочками диной ≥4. Возможно построение графа всех связей пропозиционных элементов формулы смысла и знака когнемы «Арбалет» любой наперед заданной размерности m ≤ M, однако представление его в графической форме (в виде рисунка) может оказаться затруднительным. Для M=4 такой граф будет содержать 362 цепочки, количественные сведения о которых приведены в таблице 1.

Таблица 1.

Характеристики размерности пассивного пропозиционного графа когнемы «Арбалет»

цепочки	**длина цепочки**					**всего**
	1	**2**	**3**	**4**	**5**	
старинный...->стрельба->арбалет	нет	нет	нет	1	30	**31**

оружие...->стрельба->арбалет	нет	1	2	15	123	**141**
форма...->стрельба->арбалет	нет	нет	2	4	98	**104**
лук...->стрельба->арбалет	нет	нет	1	нет	20	**21**
всего	**нет**	**1**	**5**	**20**	**271**	**297**
старинный...->орудие->арбалет	нет	нет	нет	1	21	**22**
оружие...->орудие->арбалет	нет	нет	1	2	26	**29**
форма...->орудие->арбалет	нет	нет	нет	1	9	**10**
99	нет	нет	нет	2	2	**4**
всего	**нет**	**нет**	**1**	**6**	**58**	**65**
всего	**нет**	**1**	**6**	**26**	**329**	**362**

В связи с этим построим и отобразим на рисунке 3 минимальный пассивный пропозиционный граф когнемы «Арбалет», содержащий только кратчайшие цепочки.

<<***старинный***> | <***оружие***> | <***форма***> | <***лук***>> → <***арбалет***>

Длина	Путь
4	старинный —1—> двор —1—> ствол —1—> орудие —1—> арбалет
4	старинный —1—> двор —1—> ружье —1—> стрельба —1—> арбалет
2	оружие —1—> стрельба —1—> арбалет
3	форма —1—> спорт —1—> стрельба —1—> арбалет
3	форма —1—> война —1—> стрельба —1—> арбалет
3	лук —3—> стрела —1—> стрельба —1—> арбалет
Всего цепочек 6, в том числе: 1 — нет, 2 — 1, 3 — 3, 4 — 2, 5 — нет	

Особенности построения цепочек.

Особенности получения пропозиций. Заметим, что процедура лемматизации была применена нами формально, т.е. при построении новой пропозиции мы заменили отсутствующие в АВС слова на те, которые в ней есть. Обоснование данной замены состоит в том, что она, возможно, несущественно изменила содержание формулы смысла. Большей обоснованности потребовали бы другие формальные замены, которые можно осуществить при получении поисковой пропозиции, например, взаимозамены внутри парадигмы слова, или синонимичного ряда. Приведем примеры форм с этими пропозиционными заменами.

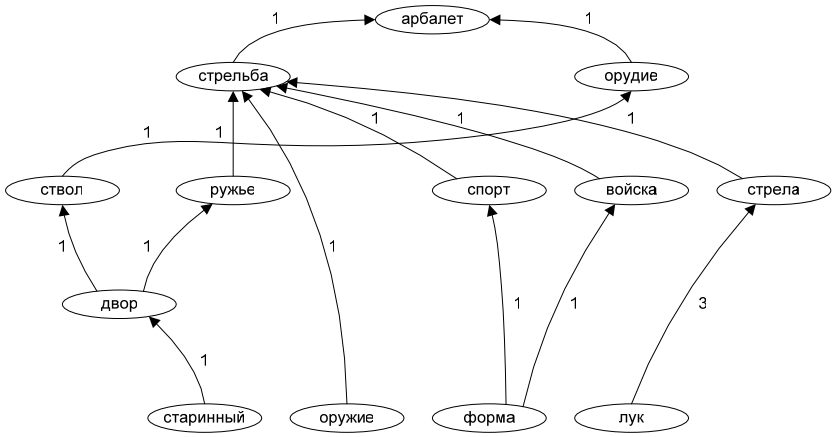


Рисунок 3. Минимальный пассивный пропозиционный граф когнемы «Арбалет»

Парадигматические формы (курсивом выделены слова-реакции, а жирным курсивом — слова-стимулы, которые есть в базе данных РАС).

Старинное ***оружие*** в форме лука [называют] арбалет.

Старинным *оружием* в форме лука [называется] арбалет.

[имя] *Старинного* *оружия* в форме лука арбалет.

Синонимичные формы.

Ряд 1: **Старый,** ветхий, *древний*, многолетний, вековой, многовековой, ***старинный***, давний, старобытный, стародавний, старомодный, устарелый, застарелый, закоснелый, закоренелый, заматерелый, давнишний, допотопный, извечный, исконный, ископаемый, архаический, археологический, престарелый, пожилой, седой, поседелый, ветеран; обветшалый, пришедший в ветхость, отживший, отсталый, затасканный, истасканный, истертый, подержанный, поношенный, потрепанный, полинялый, заскорузлый, зачерствелый.

Ряд 2: ***Старинный***, старый, дедовский, прадедовский, стародедовский, *древний*, вековой, многовековой; старосветский (уст. и книжн.); старозаветный (шутл.); антикварный (о ценной вещи).

Ряд 3: ***Форма,*** *вид*, выкройка, фасон, модель.

Ряд 4: ***Форма***, *вид*; конфигурация (книжн.).

Древнее оружие в форме лука.

Длина	Путь
4	древнее —1—> лето —2—> море —1—> война —1—> стрельба

	…
4	древнее —1—> место —1—> цепь —1—> стрела —1—> стрельба
	Всего цепочек 15
4	древнее —1—> место —1—> взять —1—> лом —1—> орудие
	…
4	древнее —1—> дело —1—> труд —1—> топор —1—> орудие
	Всего цепочек 7

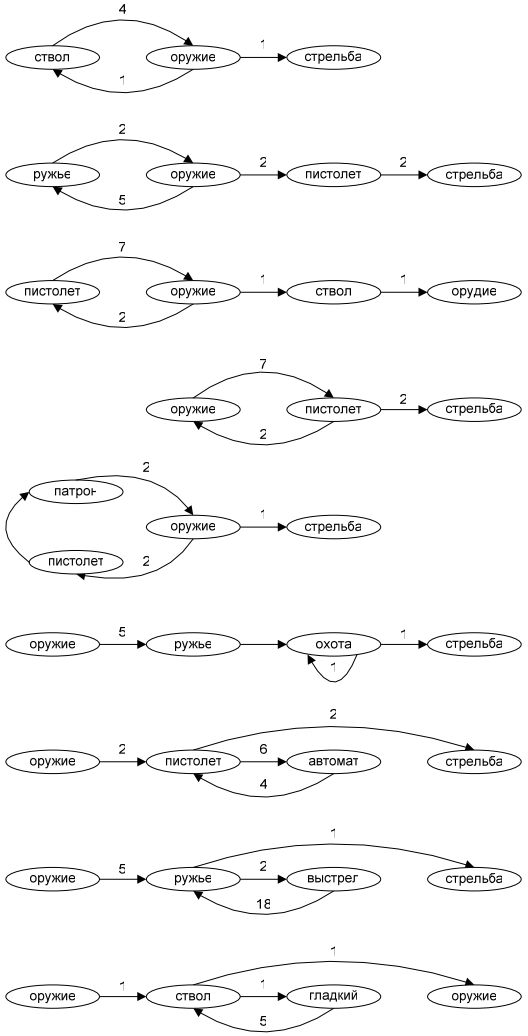
Старинное оружие в *виде* лука.

Длина	Путь
3	вид —1—> море —1—> война —1—> стрельба
3	вид —1—> окно —1—> взгляд —1—> стрельба
4	вид —1—> озеро —1—> зеркало —1—> взгляд —1—> стрельба
	…
4	вид —1—> анфас —1—> прямой —1—> стрела —1—> стрельба
	Всего цепочек 15
4	вид —1—> анфас —1—> прямой —3—> ствол —1—> орудие
	…
4	вид —1—> море —1—> след —1—> топор —1—> орудие
	Всего цепочек 5

Кольцевые элементы цепочек. Среди найденных цепочек при построении подграфа пропозиционного элемента *оружие* когнемы «Арбалет» есть 52 цепочки с закольцованными участками, например:

Путь
оружие —1—> ствол —4—> оружие —1—> стрельба
оружие —5—> ружье —2—> оружие —2—> пистолет —2—> стрельба
оружие —2—> пистолет —7—> оружие —1—> ствол —1—> орудие
оружие —2—> пистолет —7—> оружие —2—> пистолет —2—> стрельба
оружие —2—> пистолет —1—> патрон —2—> оружие —1—> стрельба
оружие —5—> ружье —5—> *охота —1—> охота —1*—> стрельба

оружие —2—> *пистолет —6—> автомат —4—> пистолет* —2—> стрельба
оружие —5—> *ружье —2—> выстрел —18—> ружье* —1—> стрельба
оружие —1—> *ствол —1—> гладкий —5—> ствол* —1—> орудие

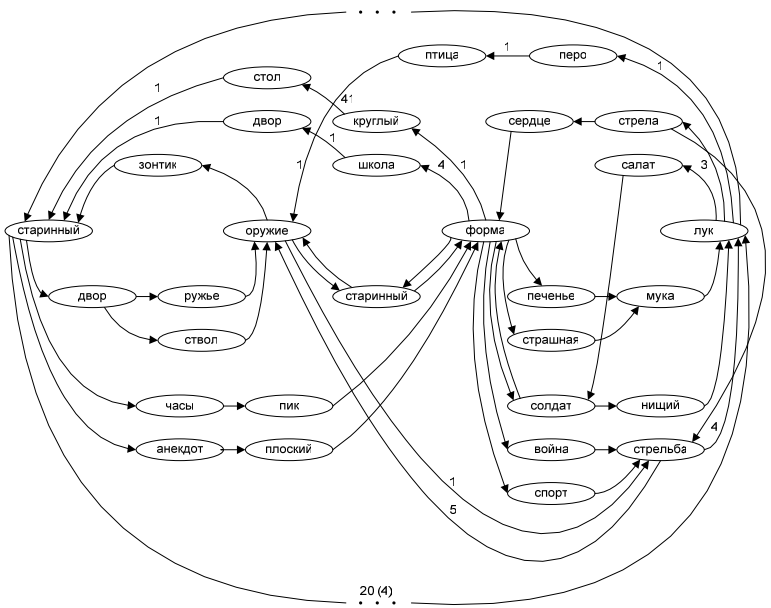


Исследование закольцованных участков цепочек требует особого внимания. Основной причиной этого является наличие в АВС значительного количества вершин модельного графа потенциально являющихся элементами колец. Этими элементами могут быть 6464 вершины типа {SR} — слова стимулы-реакции. В качестве примера приведем двух и трех элементные кольца пропозиции когнемы «Арбалет».

Длина	Путь
2	старинный —1—> вальс —1—> старинный
2	старинный —1—> двор —1—> старинный
2	старинный —10—> замок —2—> старинный
2	старинный —4—> комод —7—> старинный
3	старинный —1—> двор —1—> сквер —1—> старинный
3	старинный —1—> двор —1—> стол —1—> старинный
3	старинный —4—> комод —2—> стол —1—> старинный
3	старинный —1—> двор —1—> двор —1—> старинный
4	старинный —1—> вальс —3—> бал —2—> вальс —1—> старинный
	…
4	старинный —4—> комод —18—> старый —1—> холст —1—> старинный
	Всего колец 268, в том числе: 1 — нет, 2 — 4, 3 — 4, 4 — 262
2	оружие —2—> пистолет —7—> оружие
	…
2	оружие —1—> армия —1—> оружие
3	оружие —2—> пистолет —1—> патрон —2—> оружие
	…
3	оружие —5—> ружье —2—> выстрел —1—> оружие
4	оружие —1—> армия —1—> офицер —5—> армия —1—> оружие
	…
4	оружие —1—> ствол —1—> пушка —10—> ядро —1—> оружие
	Всего колец 262, , в том числе: 1 — нет, 2 — 5, 3 — 19, 4 — 238

2	форма —1—> армия —2—> форма
2	форма —1—> норма —2—> форма
2	форма —1—> солдат —6—> форма
3	форма —1—> армия —1—> офицер —2—> форма
	…
3	форма —1—> армия —28—> солдат —6—> форма
4	форма —1—> солдат —8—> генерал —3—> армия —2—> форма
	…
4	форма —4—> школа —1—> она —1—> юбка —1—> форма
	Всего колец 204, в том числе: 1 — нет, 2 — 3, 3 — 13, 4 — 189
2	лук —1—> салат —2—> лук
2	лук —3—> стрела —7—> лук
3	лук —1—> салат —1—> масло —1—> лук
3	лук —1—> салат —2—> капуста —1—> лук
3	лук —1—> салат —1—> салат —2—> лук
3	лук —3—> стрела —1—> Амур —2—> лук
3	лук —3—> стрела —1—> стрельба —4—> лук
4	лук —3—> стрела —1—> Амур —1—> Амур —2—> лук
	…
4	лук —3—> стрела —1—> Амур —5—> стрела —7—> лук
	Всего колец 47, в том числе: 1 — нет, 2 — 2, 3 — 5, 4 — 40

Особенности проведения ассоциативного эксперимента. При анализе получаемых цепочек, обращают на себя внимание некоторые цепочки и составляющие их стимульно-ретивные пары (звенья), имеющие в своем составе омонимичные формы — з*а*мок и зам*о*к, п*о*лон и пол*о*н, стук (настучать, стукач) и стук (звук ударов молотка), перо (стебель, лист лука), перо (средство, инструмент письма) и перо (оперенье стрелы, оконечная часть стрелы); созвучия — двор–ствол, двор–вор, забота–охота, форма–норма, салат–солдат; выражение — жрать охота. Ниже приведены примеры этих цепочек:



старинный —1—> *двор —1—> ствол* —1—> орудие
старинный —1—> *двор —7—> вор* —1—> лом —1—> орудие
старинный —10—> замок —3—> железный —3—> лом —1—> орудие
старинный —10—> замок —2—> ржавый —1—> лом —1—> орудие
старинный —1—> *стакан —1—> полон —1—> стрела* —1—> стрельба
старинный —5—> часы —1—> *забота —1—> охота* —1—> стрельба
форма —1—> норма —1—> цель —8—> мишень —5—> стрельба
лук —1—> стук —1—> жалоба —1—> спорт —1—> стрельба
лук —1—> стук —1—> молоток —10—> топор —1—> орудие
лук —1—> перо —3—> *Пушкин —1—> пушка* —1—> стрельба
лук —1—> салат —1—> солдат —2—> ружье —1—> стрельба
лук —1—> салат —1—> солдат —1—> топор —1—> орудие
лук —1—> салат —1—> *жрать —1—> охота* —1—> стрельба

Особенности взаимосвязей элементов пропозиции. Фактическим условием возможности построения пассивного пропозиционного графа какой-либо формулы смысла является отнесение составляющих ее элементов к типу слов-стимулов S или слов-стимулов-реакций SR. Из этого следует, что возможными становятся взаимосвязи между пропозиционными элементами формулы смысла. Проиллюстрируем это на примере когнемы «Арбалет», т.е. построим цепочки длиной не более 3, связывающие слова ее формулы смысла между собой.

Длина	Путь
3	лук —1—> перо —1—> птица —1—> оружие
3	лук —1—> салат —1—> солдат —6—> форма
3	лук —3—> стрела —1—> стрельба —5—> оружие
3	лук —3—> стрела —2—> сердце —1—> форма
2	оружие —1—> армия —2—> форма
2	оружие —1—> зонтик —1—> старинный
2	оружие —1—> стрельба —4—> лук
3	старинный —1—> анекдот —2—> плоский —1—> форма
3	старинный —1—> двор —1—> ружье —2—> оружие
3	старинный —1—> двор —1—> ствол —4—> оружие
3	старинный —5—> часы —4—> пик —1—> форма
2	форма —1—> армия —1—> оружие
3	форма —1—> война —1—> стрельба —4—> лук
3	форма —1—> круглый —41—> стол —1—> старинный
3	форма —1—> печенье —1—> мука —1—> лук
3	форма —1—> солдат —1—> нищий —1—> лук
3	форма —1—> спорт —1—> стрельба —4—> лук
3	форма —2—> страшная —1—> мука —1—> лук
3	форма —4—> школа —1—> двор —1—> старинный
	Всего цепочек 19

Общее количество таких цепочек оказалось равным 19. При этом не нашлось ни одной цепочки длиной 3, которая бы связывала слова *старинный* и *лук*. Однако есть 20 цепочек длиной 4 *<старинный —…—> лук>* и 28 — *<лук —…—> старинный>*.

Другие примеры.

Приведем несколько примеров пассивных пропозиционных графов размерности 3 для когнем: «Ряска», «Раскопки» и «Серп».

Пример 1. Знак = *Ряска.* Формула смысла = *Зеленое одеяло водоемов*.

Пропозиция формулы смысла = *<зеленый>* | *<одеяло>* | *<водоем>*.

Длина	Кратчайшие пути не более 4
3	зеленый —1—> лягушка —7—> болото —1—> ряска
3	зеленый —1—> газ —1—> болото —1—> ряска
3	зеленый —10—> крокодил —3—> болото —1—> ряска
3	зеленый —1—> лягушка —2—> пруд —1—> ряска
4	одеяло —1—> ночь —1—> туман —1—> болото —1—> ряска
4	одеяло —1—> ночь —1—> рассвет —1—> болото —1—> ряска
3	водоем —3—> лягушка —7—> болото —1—> ряска
3	водоем —1—> море —1—> болото —1—> ряска
3	водоем —10—> озеро —2—> болото —1—> ряска
3	водоем —2—> река —1—> болото —1—> ряска
3	водоем —6—> вода —1—> болото —1—> ряска
3	водоем —3—> лягушка —2—> пруд —1—> ряска
3	водоем —2—> река —1—> пруд —1—> ряска
3	водоем —1—> речка —1—> пруд —1—> ряска
3	водоем —4—> рыба —1—> пруд —1—> ряска
	Всего цепочек 15

Пример 2. Знак = *Раскопки.* Формула смысла = *Повседневная работа археолога*.

Пропозиция формулы смысла = *<повседневный>* | *<работа>* | *<археолог>*.

Длина	Кратчайшие пути не более 3
3	повседневный —1—> обед —1—> редкость —1—> раскопки
3	работа —1—> клад —1—> древний —1—> раскопки
	Всего цепочек 2

Пример 3. Знак = *Серп.* Формула смысла = *Что держит в руках Мухинская колхозница*.

Пропозиция формулы смысла = *<держать>* | *<рука>* | *<Мухина>* | *<колхозница>*.

Длина	Кратчайшие пути не более 3
3	держать —1—> руль —2—> золотой —1—> серп
3	держать —3—> руки —1—> нож —1—> серп
3	рука —1—> удар —1—> молот —32—> серп
3	рука —2—> перстень —14—> золотой —1—> серп
3	рука —2—> голова —1—> коса —1—> серп
3	рука —1—> красота —1—> коса —1—> серп
3	рука —1—> работа —2—> молоток —6—> серп
3	рука —1—> удар —1—> молоток —6—> серп
3	рука —2—> палец —1—> нож —1—> серп
3	рука —2—> река —1—> трава —1—> серп
3	рука —1—> гнев —1—> трава —1—> серп
	Всего цепочек 11

Активный режим работы когнайзера.

Фактической целью рассмотрения однонаправленных цепочек было моделирование перехода от смысла (пропозиции формулы смысла) к знаку. Теперь изменим цель. Попытаемся перейти от знака к смыслу, т.е. покажем на примере, что возможно решение и обратной задачи —

для заданного знака построение такой пропозиции, которая эквивалентна априори известному смыслу (пропозиции формулы смысла).

На рис. 4 приведен активный пропозиционный граф когнемы «Арбалет», цепочками которого являются:

Длина	Путь
1	арбалет <—1— орудие
1	арбалет <—1— стрельба
	Всего цепочек 2, в том числе: 1 — 2

Длина	Путь
	Орудие - лук
3	орудие —1—> работа —2—> мука —1—> лук
3	орудие —4—> труд —1—> мука —1—> лук
3	орудие —1—> выстрел —1—> стрела —7—> лук
3	орудие —1—> пистолет —2—> стрельба —4—> лук
3	орудие —4—> пушка —1—> стрельба —4—> лук
3	орудие —1—> топор —1—> капуста —1—> лук
4	орудие —1—> выстрел —1—> стрела —1—> Амур —2—> лук
4	орудие —4—> труд —1—> морковь —1—> капуста —1—> лук
	…
4	орудие —8—> убийство —1—> пистолет —2—> стрельба —4—> лук
	Всего цепочек 89, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 6, 5 — 83
	Орудие - оружие
2	орудие —1—> пистолет —7—> оружие
	…
2	орудие —1—> выстрел —1—> оружие
3	орудие —1—> пистолет —1—> патрон —2—> оружие
	…
3	орудие —1—> выстрел —1—> выстрел —1—> оружие

4	орудие —1—> топор —3—> дерево —1—> армия —1—> оружие
	...
4	орудие —1—> выстрел —4—> пушка —10—> ядро —1—> оружие
	Всего цепочек 271, в том числе: 1 — нет, 2 — нет, 3 — 4, 4 — 17, 5 — 250
	Орудие - старинный
3	орудие —1—> работа —1—> стол —1—> старинный
4	орудие —1—> выстрел —3—> последний —1—> вальс —1—> старинный
	...
4	орудие —1—> работа —1—> творить —1—> холст —1—> старинный
	Всего цепочек 109, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 1, 5 — 108
	Орудие - форма
3	орудие —1—> работа —1—> фигура —1—> форма
	...
3	орудие —4—> пушка —1—> солдат —6—> форма
4	орудие —1—> топор —3—> дерево —1—> армия —2—> форма
	...
4	орудие —2—> лопата —1—> новая —1—> юбка —1—> форма
	Всего цепочек 132, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 6, 5 — 126

Длина	Путь
	Стрельба - лук
1	стрельба —4—> лук
3	стрельба —4—> мишень —1—> стрела —7—> лук
3	стрельба —7—> пистолет —2—> стрельба —4—> лук
3	стрельба —4—> ружье —1—> стрельба —4—> лук
3	стрельба —3—> война —1—> стрельба —4—> лук
3	стрельба —4—> мишень —5—> стрельба —4—> лук
3	стрельба —5—> оружие —1—> стрельба —4—> лук
3	стрельба —1—> заяц —1—> капуста —1—> лук

4	стрельба —4—> мишень —1—> стрела —1—> Амур —2—> лук
	…
4	стрельба —4—> ружье —3—> пушка —1—> стрельба —4—> лук
	Всего цепочек 61, в том числе: 1 — нет, 2 — 1, 3 — нет, 4 — 7, 5 — 53
	Стрельба - оружие
1	стрельба —5—> оружие
2	стрельба —7—> пистолет —7—> оружие
2	стрельба —4—> ружье —2—> оружие
2	стрельба —2—> убийство —2—> оружие
3	стрельба —7—> пистолет —1—> патрон —2—> оружие
3	стрельба —1—> пулемет —1—> патрон —2—> оружие
	…
3	стрельба —4—> мишень —4—> выстрел —1—> оружие
4	стрельба —2—> убийство —1—> тюрьма —1—> армия —1—> оружие
4	стрельба —1—> огонь —1—> дерево —1—> армия —1—> оружие
	…
4	стрельба —4—> ружье —3—> пушка —10—> ядро —1—> оружие
	Всего цепочек 238, в том числе: 1 — нет, 2 — 1, 3 — 3, 4 — 27, 5 — 207
	Стрельба - старинный
3	стрельба —5—> оружие —1—> зонтик —1—> старинный
3	стрельба —1—> огонь —7—> камин —2—> старинный
4	стрельба —3—> война —1—> окно —2—> двор —1—> старинный
4	стрельба —1—> заяц —1—> снег —1—> двор —1—> старинный
	…
4	стрельба —1—> огонь —1—> уголь —1—> холст —1—> старинный
	Всего цепочек 65, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 2, 5 — 63
	Стрельба - форма
3	стрельба —5—> оружие —1—> армия —2—> форма
3	стрельба —7—> пистолет —1—> милиция —2—> форма

3	стрельба —1—> кровь —3—> сдать —1—> форма
3	стрельба —1—> огонь —1—> сердце —1—> форма
4	стрельба —2—> убийство —1—> тюрьма —1—> армия —2—> форма
4	стрельба —1—> огонь —1—> дерево —1—> армия —2—> форма
	…
4	стрельба —1—> кровь —1—> новая —1—> юбка —1—> форма
	Всего цепочек 90, в том числе: 1 — нет, 2 — нет, 3 — нет, 4 — 4, 5 — 83

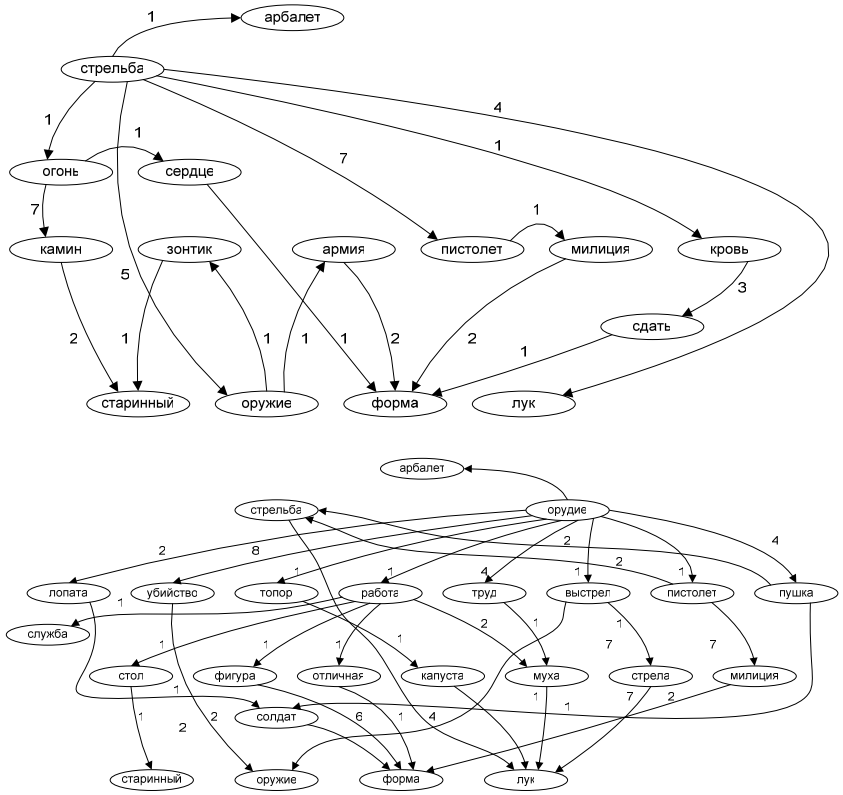


Рисунок 4. Минимальные активные пропозиционные подграфы размерности 4 когнемы «Арбалет»

Алгоритмы поиска путей в ассоциативной сети.

Рассмотренная графовая модель АВС и поиск ассоциативных цепочек относятся к классу задач поиска связей между вершинами в графах, которые давно известны и хорошо освоены как теоретически, так и практически. Задача состоит в определении эффективной последовательности просмотра (обхода) вершин графа. Описано множество различных алгоритмов ее принципиального решения, а также модификаций, позволяющих учесть многие особенности графов и целей поиска. Наиболее известными подходами являются «поиск в глубину» и «поиск в ширину». В числе самых известных алгоритмов следует назвать: «волновой», Форда-Беллмана, Флойда, Дейкстры. Данные алгоритмы используются для решения поисковой задачи при различных начальных условиях: например, граф может быть ориентированным или нет, взвешенным или нет, требуется найти кратчайшие или все пути, есть ли в графе циклические участки и т.д. Приведем их краткие описания.

Альтернативное осознавание.

Альтернативные варианты моделирования осознавания, так названные в начале данного раздела, фактически представляют собой некоторую процедуру принятия решений. В этом случае функционирование когнайзера состояло бы в преобразовании исходных данных в результирующие: исходными данными для него являлись бы либо ЕЗМ, например, в форме пропозиции ВФС какой-либо когнемы, либо ЯЕ; а результатом — выбранная из некоторого множества конкретная ЯЕ, или сформированная (созданная, сгенерированная) ЕЗМ, соответственно.

В основу архитектуры (рис. 5) такой модели когнайзера может быть положена традиционная 2–3-х компонентная структура, включающая: *Базу лингвистических данных* (БЛД) и *Лингвистический процессор* (ЛП), состоящий из *Интерфейса* и *Решателя*.

Отметим ряд особенностей и требований, которые следует учесть при моделировании.

1. При моделировании безальтернативного осознавания (рассмотренные пассивный и активный режимы работы когнайзера с априори известными знаком и формулой смысла) использовались только частично база данных ассоциативного эксперимента и две компоненты когнем базы данных когнитивного эксперимента. При моделировании альтернативного осознавания должны быть найдены методы учета значений других компонент когнем — способа задания смысла, референтной области и функции; а также частотных (валентных) характеристик стимульно-реактивных пар.

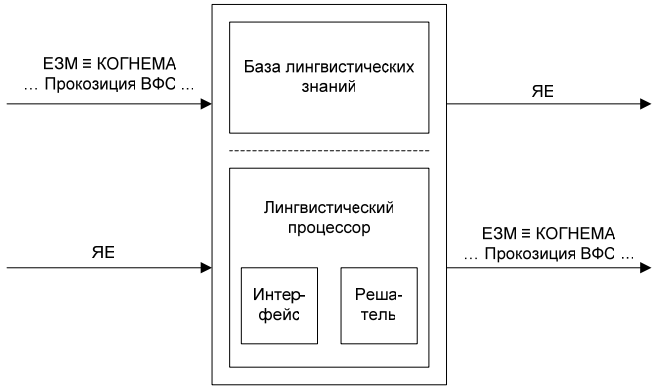


Рисунок 5. Архитектура модели когнайзера.

2. Особого внимания заслуживают такие компоненты когнем как формула смысла и референтная область. В отличие от других компонент когнем, которые представлены простыми списками, они имеют более сложную структуру. Так формула смысла в большинстве случаев представлена предложениями различного типа, понимание которых возможно только в определенном виртуальном контексте, или как элемент некоторого гипертекста. Референтная область, представлена в когнеме наименованием одной или нескольких подобластей, отношения между которыми являются иерархическими или сетевыми, в общем случае относятся к типу «многие-ко-многим».

3. БЛД должна включать не только данные ассоциативного и когнитивного экспериментов, но и другие данные, позволяющие «сужать» и «расширять» множества альтернатив ЯЕ и ЕЗМ. К таким данным следует относить сведения о: правилах словообразования и морфемном членении слов, синонимах, антонимах, фразеологизмах, метафорических конструкциях, терминах и др. Эти сведения сосредоточены в различных лексикографических источниках. Такие требования кардинально изменяют представление о БЛД. В случае даже их частичного удовлетворения получается интегрированная лексикографическая (словарно-тезаурусная) система.

4. Лингвистический процессор модели когнайзера (его Решатель) реализует две укрупненные функции: поиск (отбор) альтернативных ЕЗМ или ЯЕ и последующий выбор одной из них. Существует множество формальных интерпретаций этих функций в зависимости от цели и предмета поиска. Важнейшими характеристиками при этом являются

критерии отбора и выбора альтернатив. Основные особенности реализации поискового ЛП когнайзера видятся: а) в его вариативности, т.е. использовании поликритериальных оценок альтернатив; б) в вероятностном (неточном) оценивании, как самих альтернатив, так и процесса поиска; в) в нечеткости (степени достоверности) критериев поиска и его результатов.

5. Интерфейс ЛП обеспечивает внутреннюю взаимосвязь компонентов когнайзера и внешние взаимодействия. Основной особенностью внешних взаимодействий является ориентация на ученого-исследователя языкового сознания. Интерфейс когнайзера по своей форме является визуальным (графическим) образом модели языкового сознания, оснащенным своеобразным «пультом управления». Основными функциями управления процессом моделирования являются: манипулирование ЕЗМ и ЯЕ — создание, изменение, выбор и т.п.; пуск/приостановка/остановка моделирования; создание и манипулирование БЛД; установка и изменения условий моделирования; визуализация процесса моделирования и его результатов; документирование результатов моделирования и др.

Заключение

В редакционной статье сборника ИТС8 отмечалось о важности для анализа результатов когнитивных экспериментов, целью которых является моделирование вербального сознания, использование в качестве механизма интерпретации формальных моделей различного типа, в том числе алгебраических. В этой статье такими моделями являлись «интуитивно понятные» графовые модели АТ.

Для алгебраического описания АТ введем понятие ассоциативного выражения (АВ) — класса синкопированных алгебраических выражений, состоящих из слов-стимулов (S_i) и слов-реакций (R_j), в общем случае ЯЕ ассоциативного тезауруса, связанных между собой обозначениями арифметических, логических операций и кванторов, а также операций ассоциирования: «собирания» реакций на стимулы ($S_i \rightarrow \{R_j\}$) и «восстановления» стимулов на реакции ($R_j \leftarrow \{S_i\}$).

Пара <ВФС→Зн> изначально может быть представлена в форме ассоциативных компонент — ЯЕ ассоциативного тезауруса, являющихся эквивалентами Зн и составляющих ВФС слов и словосочетаний, а затем, в форме ассоциативного выражения. Такое представление позволяет: во-первых, описать основную составляющую когнемы как часть АТ и смоделировать пассивный режим работы ЯС как активный; во-вторых, инкапсулировать когнемы в АТ, т.е. представить его в виде проекции

когнитивного тезауруса, и тем самым смоделировать активный режим работы ЯС как пассивный.

АВ ВФС, инкапсулированное в АТ, представляет собой множество цепочек, состоящих из стимулов и реакций, связывающих отношениями ассоциирования компоненты ВФС и Зн. Длина цепочки, между узлом «слово-знак» и компонентом ВФС является ассоциативным расстоянием и определяется количеством отношений в ней. В общем случае для конкретной пары <ВФС→Зн> в АТ может быть выявлено множество АВ (графов), основными параметрами которых будут являться значения минимального и максимального ассоциативных расстояний, количество стимулов и реакций (узлов графа), а также цепочек.

Одной из важных задач инкапсуляции является нахождение цепочек заданной и минимальной длины, т.е. АВ с заданными параметрами. Основной операцией при этом является *свертка*, под которой понимается последовательное применение правил эквивалентного преобразования ассоциативного выражения, а также его конечная форма, получаемая в результате их применения. Целью свертки является получение минимальной формы записи АВ ВФС некоторого Зн.

Дальнейшие суждения в данном направлении, формально описанные в нотации алгебр, могут привести к разработки специфических ассоциативных или когнитивных алгебраических моделей языкового сознания.

Литература

Караулов 2003а *Караулов Ю.Н.* Единицы языка и единицы знания. // Теория и практика лингвистического анализа текстов СМИ в судебных экспертизах и информационных спорах: Материалы научно-практического семинара. Ч. 2. М., 2003 а.

Караулов 2003б *Караулов Ю.Н.* Языковое сознание: пассивный и активный режим работы // Межкультурная коммуникация и перевод: Материалы межвузовской научной конференции 31 января 2003 г. М., 2003 б.

Караулов 2004а *Караулов Ю.Н.* Концептография языковой картины мира. Статья 1. Первый этап ”восхождения” к образу мира: от элементарных фигур знания к предметно-референтным областям культуры // Проблемы прикладной лингвистики. Вып. 2. М., 2004

Караулов 2004б *Караулов Ю.Н.* Концептография языковой картины мира. Статья 2. Референтные области, концепты и концептосферы. (Второй этап “восхождения” – от областей к концептам). // Языковое сознание: теоретические и прикладные аспекты. Москва – Барнаул, 2004

Караулов 2004в *Караулов Ю.Н.* Основы лингвокультурного тезауруса русского языка. // Русское слово в русском мире. Калуга, Изд. “Эйдос”, 2004

РАС 2002 Русский ассоциативный словарь. В 2 т. Т. 1. От стимула к реакции: Ок. 7000 стимулов / Ю.Н.Караулов, Г.А.Черкасова, Н.В.Уфимцева, Ю.А.Сорокин, Е.Ф.Тарасов. – М.: «Астрель»: ООО «АСТ», 2002. – 784 с.

Статьи аспирантов и студентов

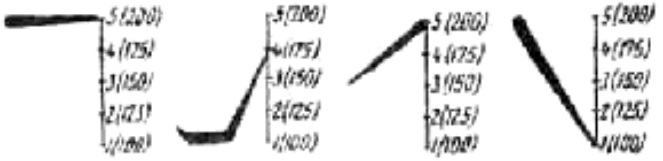
Ван Чэнь

ПРЕДСТАВЛЕНИЕ КИТАЙСКИХ ИЕРОГЛИФОВ В КОМПЬЮТЕРНОЙ СРЕДЕ

Система полных тонов в китайском языке и их смыслоразличительная функция

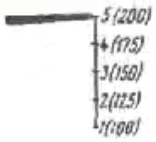
Слоги в китайском языке отличаются не только своим звуковым составом (согласными и гласными), но и тоном, или мелодией. Каждый слог, получающий ударение (сильное или хотя бы слабое), произносится тем или иным тоном. В китайском общенародном языке Путунхуа, базирующемся на пекинском диалекте, имеются четыре полных тона, каждый из которых характеризуется совокупностью присущих ему качеств. Для различения смысла важны и тон, и звуковой состав слога; одно и то же сочетание звуков передает совершенно разные значения в зависимости от того, каким тоном оно произнесено. Какова же характеристика каждого из четырех полных тонов? Каждый из полных тонов китайского языка характеризуется совокупностью определенных признаков: 1) направлением движения основного тона (формой тона), 2) распределением интенсивности (силы звука) внутри тона, 3) частотным диапазоном (высотным интервалом между начальной и конечной точками тона), 4) высотой тона, 5) временем звучания (долготой тона).

Мелодический рисунок четырех тонов можно графически изобразить следующим образом:



Вертикальная черта с цифрами 1-5 представляет собой общепринятую шкалу, условно обозначающую общий диапазон речевого голоса, охватывающий четыре тона (для мужского голоса этот диапазон простирается в среднем от 100 до 200 герц).

Высота тона определяется звуковой частотой в герцах. Толстая черта условно обозначает форму тона, а различная толщина этой линии показывает распределение интенсивности внутри тона. Сравним следующие примеры:

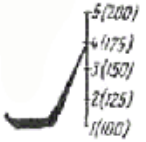


мā (ma) - мать
шу (shu) - книга
да (da) - строить

Мелодия **первого тона** - высокая, ровная, долгая, с равномерной интенсивностью и лишь некоторым ослаблением ее к концу (на русского слушателя производит впечатление незаконченного высказывания).

мá (ma) - конопля
шу (shu) - спелый
да (da) - отвечать

Мелодия **второго тона** - краткая, быстро восходящая (с диапазоном 140-200 герц), с максимумом интенсивности в конце слога производит впечатление переспроса).



мă (ma) - лошадь
шу (shu) - считать
да (da) - бить

Третий тон - низкий, долгий, имеет нисходяще-восходящую форму (с диапазоном 120-100-180 герц), с максимумом интенсивности на низкой ноте (производит впечатление недоуменного вопроса).

мà (ma) - ругать
шу (shu) - дерево
да (da) - большой

Четвертый тон - краткий, резко нисходящий от высшей точки до низшей (с диапазоном 200-100 герц). Падение тона сопровождается рез-

ким ослаблением интенсивности (мелодия четвертого тона производит впечатление категорического приказания).

В китайском языке нередки и такие случаи, где даже двусложные слова различаются только тонами. Такие слова можно разделить на три группы.

а) слова, отличающиеся тоном конечного слога:

б) слова, отличающиеся тоном начального слога:

цзяоши (jiaoshi) - преподаватель

цзяоши (jiaoshi) - аудитория

суншу (songshu) - белка
суншу (songshu) - сосна

бинъюань (bingyuan) - больной
бинъюань (bingyuan) - больница

тунши (tongshi) - одновременно
тунши (tongshi) - коллега

туйхуань (tuihuan) - возвратить
туйхуань (tuihuan) - обменять (купленную вещь)

сюгай (xiugai) - строить
сюгай (xiugai) - исправить

шупи (shupi) - обложка книги
шупи (shupi) - кора дерева

тухуа (tuhua)- рисунок
тухуа (tuhua) - местный говор

бинчуан (bingchuang) - санки
бинчуан (bingchuang) - больничная койка

нули (nuli) - раб
нули (nuli) - усердный

беймянь (beimian) - северная сторона
беймянь (beimian) - задняя сторона, тыл

дахуа (dahua) - ответ
дахуа (dahua) - хвастовство

в) слова, отличающиеся тонами обоих слогов:

цывэй (ciwei) - суффикс
цывэй (ciwei) - еж

эюй (eyu) - русский язык
эюй (eyu) - крокодил

баовэй (baowei) - окружать
баовэй (baowei) - защищать

тунъи (tongyi) - одинаковый
тунъи (tongyi) - согласие
тунъи (tongyi) - единство

тончжи (tongzhi) - объявление
тончжи (tongzhi) - товарищ
тончжи (tongzhi) - господствовать

Существенные и дополнительные признаки тонов. Как же отличаются тоны друг от друга, если они обладают пятью перечисленными признаками? Прежде всего, не все пять признаков одинаково важны для характеристики тона - есть признаки *существенные* (направление движения основного тона, распределение интенсивности и высотный интервал между начальной и конечной точками тона) и *дополнительные* (высота тона и время звучания, или долгота тона). Дополнительные признаки проявляются только в сопоставлении тонов друг с другом и могут варьировать в ту или другую сторону при произнесении отдельно взятого слога, не влияя при этом на качественную характеристику тона. Возьмем *долготу тона.* По этому признаку различие между тонами идет в следующем порядке: третий тон характеризуется наибольшим временем звучания; первый тон, хотя и называется долгим, но по сравнению с третьим он заметно короче; второй же тон несколько короче первого; наконец, самый короткий - четвертый тон. Если, например, время произнесения слога в третьем тоне равняется 500 мсек2, то время звучания первого тона соответственно равно приблизительно 400 мсек, второго тона - 350-375 мсек, а четвертого тона -примерно 200-225 мсек. Но эти данные не являются постоянными, они могут в значительной мере изменяться в зависимости от тех или иных условий: от общего темпа речи, от ее эмоциональной окраски, от фразового ударения. А если говорить о тоне отдельно взятого слога (без сопоставления с другими тонами), то время его звучания реализуется говорящим произ-

вольно. Такого же рода признаком является и *высота тона.* Высота тона отдельно взятого слога также реализуется говорящим произвольно. Первый тон, например, может быть произнесен и на высоте 200 герц, и 300 герц, и, наоборот, ниже 200 герц, - но это нисколько не влияет на качественную характеристику тона. В связной речи высота тонов зависит прежде всего от диапазона речевого голоса говорящего, а также от фразового ударения и от эмоциональных факторов. В отличие от этих двух признаков другие *три признака (существенные)* присущи в неизменном виде каждому данному тону в любом случае, как при сопоставлении тонов друг с другом, так и при изолированном произнесении слога. Именно они составляют ту совокупность качеств, которая и позволяет воспринимать тон как таковой. Рассмотрим каждый из этих трех признаков.

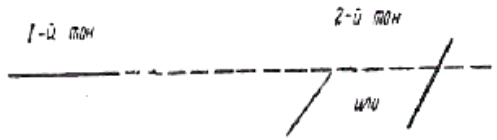
По направлению движения основного тона первый тон характеризуется как ровный, второй -восходящий, третий - нисходяще-восходящий, четвертый - нисходящий. Направление движения основного тона является наиболее существенным признаком тона. Даже небольшие изменения в движении тона могут привести к тому, что на слух будет воспринят совсем другой тон. Например, если в первом тоне ровная мелодия не будет выдержана до самого конца звучания слога и в конце слога будет снижена (как это нередко наблюдается у учащихся, овладевающих китайскими тонами), то первый тон в этом случае будет похож на четвертый. Наиболее сложной по форме является мелодия третьего тона. Как видно из графического изображения, она складывается из трех частей: нисходящей, ровной и восходящей. При этом существенными являются лишь две последние части, на которые приходится приблизительно четыре пятых времени звучания слога. Начальная же часть, ввиду ее кратковременности, слабо воспринимается на слух, а иногда и вовсе не улавливается.

По распределению интенсивности (усилению и ослаблению голоса) тоны различаются следующим образом: первый тон характеризуется равномерной интенсивностью; для второго тона характерны сравнительно слабое начало и заметный подъем интенсивности к концу слога; третий тон имеет максимум интенсивности в начале, на низкой ноте, с заметным ослаблением к концу слога; в четвертом тоне наиболее интенсивно начало слога, затем резко нисходящая мелодия сопровождается таким же резким падением интенсивности. Таким образом, по направлению движения тона и распределению интенсивности взаимно противоположны, например второй и четвертый тоны: второй тон с восходящей мелодией и усилением интенсивности к концу слога и четвертый

тон с нисходящей мелодией и ослаблением интенсивности к концу слога. Если сопоставить второй и третий тоны, то мелодия третьего тона не просто восходящая, а с ровным началом и восходящим концом. Своей восходящей частью, на которую приходится большая часть времени звучания слога, третий тон напоминает второй. При таких условиях важное значение приобретает другой фактор-распределение интенсивности: во втором тоне максимум интенсивности приходится на конец слога, а в третьем-на начало. Если же восходящая мелодия второго тона не будет сопровождаться усилением голоса к концу слога (что часто допускают учащиеся), то такой тон будет похож на третий. И наоборот, если в третьем тоне начало произносится недостаточно интенсивно и к концу слога голос не ослабевает или ослабевает недостаточно, такой тон оказывается похожим на второй. Этим же объясняется и тот факт, что второй тон, произнесенный китайцем довольно низко, или третий тон, произнесенный довольно высоко, удается определить иногда только по месту сосредоточения интенсивности.

Высотный интервал между начальной и конечной точками тона. Соблюдение интервалов между начальной и конечной точками тона совершенно обязательно, независимо от того, на какой высоте произносится тон. Однако в практике преподавания китайских тонов часто приходится наблюдать случаи, когда учащиеся не выдерживают полного диапазона четвертого тона, не доводят этот тон до необходимой нижней точки, и при быстром произнесении он оказывается похожим на первый тон. Особо следует сказать о высотном интервале второго тона. На приведенном графическом изображении мелодии тонов показано, что конечная точка второго тона доходит до высоты, на которой находится первый тон. Но существует и другой, не менее распространенный вариант второго тона - его конечная точка заметно выходит за пределы высоты первого тона, т. е. условной высоты.

Графически это можно изобразить так:



Такова общая характеристика четырех тонов, каждого в отдельности и в сопоставлении друг с другом.

Распределение слогов путунхуа по тонам. Слог и морфема. Известно, что система путунхуа содержит около 400 слогов, различающихся по звуковому составу. Наличие тонов умножает это количество. Но было бы неправильно считать, что каждый слог китайского языка может быть произнесен четырьмя тонами, давая при этом соответствующие морфемы (знаменательные или, реже, служебные). Далеко не каждый звуковой состав слога представлен во всех четырех тонах. Лишь около половины общего количества слогов (174) имеет по четыре тоновых варианта; несколько меньшее количество слогов (148) имеет по три тоновых варианта; 57 слогов представлены в двух тонах; 25 слогов существуют только в одном тоне.

Иероглифическая письменность Китая

В китайском письме более 40,000 символов, но всего 400 слогов, каждый из которых может иметь по 4 тона. В 1960-е гг. в КНР была проведена реформа иероглифики, количество черт практически во всех знаках значительно сократилось. Однако китайцы Тайваня, Гонконга, Аомыня, Сингапура нововведение не приняли.

Основных правил написания китайских иероглифов - семь:

1. сначала пишется горизонтальная черта, после - вертикальная;
2. сначала пишется откидная влево, затем - откидная вправо;
3. иероглиф пишется сверху вниз;
4. ...и слева направо;
5. в первую очередь, пишется внешняя часть иероглифа, затем - то, что внутри;
6. в таких иероглифах, как, скажем, "государство", "день, солнце" сначала пишутся элементы внутри, в завершении же он "запечатывается" снизу;
7. вначале следует писать тот, элемент иероглифа, что посередине, в последнюю очередь - элементы слева и справа соответственно.

Способы ввода китайских иероглифов

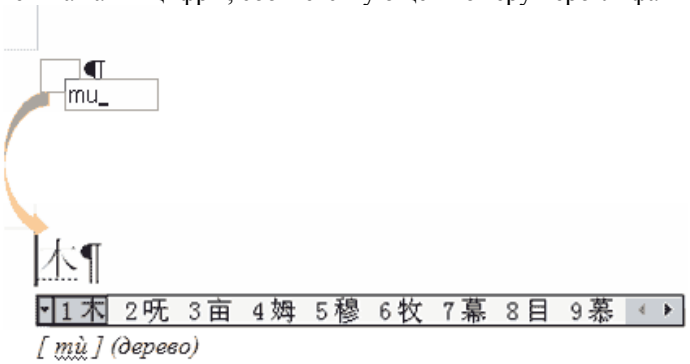
1. Ввод в транскрипции пиньинь (по звучанию). Пиньинь – это разновидность транскрипции, позволяющая передавать на письме звучание китайских иероглифов и слов латинскими буквами с указанием тона. Для ввода иероглифов в транскрипции пиньинь, достаточно иметь стандартную клавиатуру с латинской раскладкой.

Введите транскрипцию слова, затем, также с помощью клавиатуры, укажите один из четырех тонов или наберите точку, если тон отсутствует.

Обозначение тона:

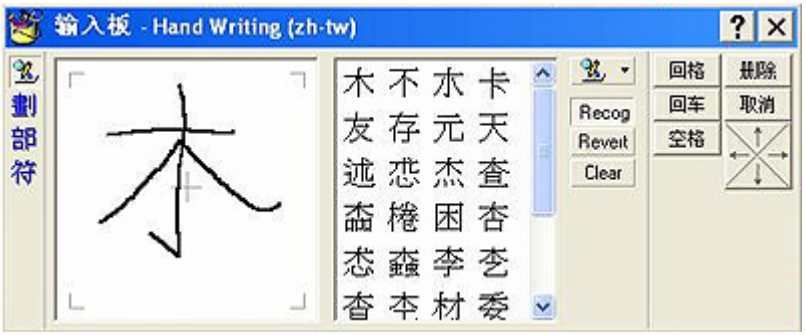
1 2 3 4 (нет тона)

Появится меню со списком всех иероглифов, имеющих указанное звучание. Выбрать из них нужный можно при помощи мышки или путём нажатия цифры, соответствующей номеру иероглифа в меню.



Обратите внимание: одной записи в транскрипции пиньинь может соответствовать до 25 различных иероглифов, имеющих совершенно разный смысл.

2. Ввод иероглифов в графическом режиме. При этом способе, удобном в тех случаях, когда написание вы помните лучше произношения, следует схематично изобразить иероглиф в специальном поле. Рядом с полем будут представлены варианты иероглифов с похожим написанием.



3. Ввод с использованием фонетического алфавита Чжуинь. Для ввода символов алфавита Чжуинь следует использовать виртуальную клавиатуру. Сначала вводят фонетические символы, затем указывают нужный тон. После этого появляется возможность выбрать нужный иероглиф из всех, имеющих указанное звучание.

Гуань Ли, Ван Чэнь

ПРОГРАММА АВТОМАТИЧЕСКОГО ПЕРЕВОДА КИТАЙСКИХ МЕТАФОР ИНФОРМАТИКИ

О программе

В настоящее время для непрерывного поступательного развития науки и техники каждой страны должен происходить научно-технический и культурный обмен знаниями и идеями. Для этого нужно знать интернациональный язык общения. Сегодня этим языком является английский. Но не только на нем выходят научные публикации. Есть страны, ученые которых публикуют свои работы в разных предметных областях на языке своей страны. Россия и Китай являются примерами таких стран. Значительная часть научных публикаций и в России и в Китае выходит на русском и китайском языках соответственно. Для развития научно-технического и культурного российско-китайского сотрудничества важно эффективное решение проблемы перевода научных публикаций. Одним из путей в данном направлении является использование систем автоматического перевода (АП) — компьютерных программ и электронных словарей.

Целью работы, основная компонента которой представлена в настоящей статье, является создание макета словаря метафор в предметной области «информатика и вычислительная техника» для целей автоматического китайско-русского перевода.

В работе поставлены и решаются следующие задачи:

- исследование особенностей китайско-русского перевода технических текстов в области информатики и вычислительной техники;
- исследование использование метафор (тропов) в текстах по информатике и вычислительной технике на китайском и русском языках;
- анализ электронных словарей и программ автоматического перевода, поддерживающих китайско-русский перевод;
- создание программ автоматического перевода метафор для перевода текстов в области информатики и вычислительной техники.

При решении поставленных задач были проанализированы статьи общеизвестных компьютерных журналов, и журналов, издаваемых

только в Китае. Анализировались статьи, написанные только носителями языка.

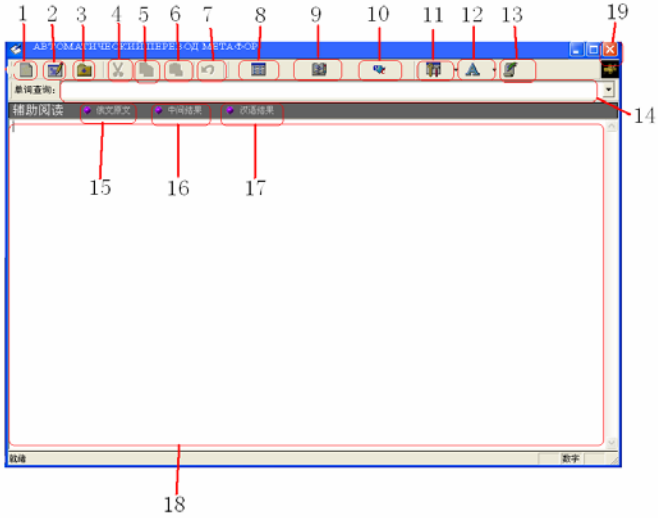
Были исследованы принципы работы программ перевода, выявлены их слабые и сильные стороны и различия в «восприятии» текста машиной и человеком. В результате анализа словарей и программ АП был сделан вывод о необходимости применения словаря метафор для улучшения качества АП.

В программу автоматического перевода метафор были включены наиболее распространенные тропы, встречавшиеся при переводе текстов. Описание содержания словарной компоненты программы, собственно «Словаря китайских метафор информатики», представлено в ранее опубликованной статье [1].

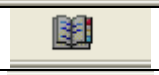
Программа «Автоматический перевод метафор» написана в среде программирования Delphi. Она записана на компакт-диск «Автоматический перевод метафор», а для ее выполнения необходимо запустить файл page.exe. После запуска программы появится заставка, а затем главное окно.

Работа в программе

Главное окно программы состоит из основной формы:

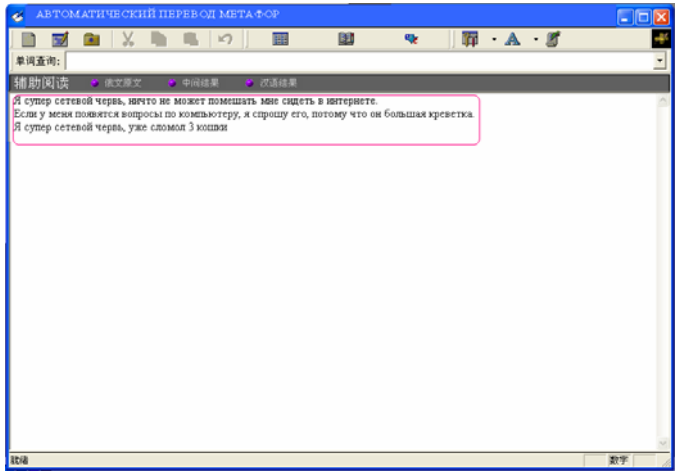


№	Изображение	Назначение
1		связь с окном создания документа

2		связь с окном открытия документа
3		связь с окном сохранеия документа
4		вырезается текст
5		копируется текст
6		вставляется текст
7		отмена ввода
8		связь с окном “О ПРОГРАММЕ”
9		быстрый словарь
10		быстрый автоматический перевод метафор
11		связь с окном регистрации новых слов и изменения свойств слов
12		размер шрифта
13		связь с окном русской клавиатуры
14		окно словаря
15		связь с окном русского текста
16		связь с окном автоматического перевода метафор
17		связь с окном китайского результата перевода метафор
18		окно автоматического перевода метафор
19		Отмена

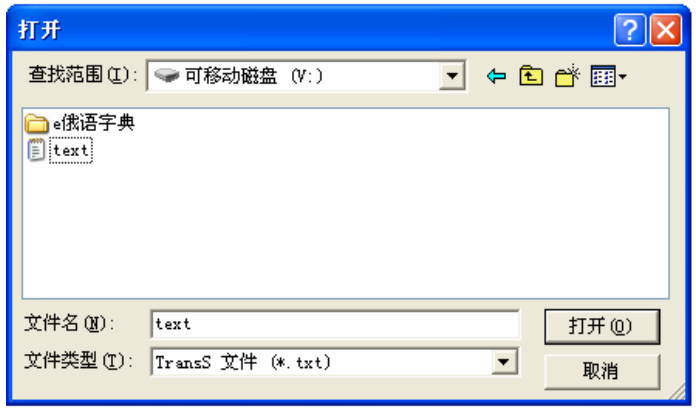
Работа с созданием документа:

Для создания документа используется кнопка  , нажав которую вы получаете новый документ в основной колонке, чтобы записать русский текст в колонке, как это показано на рис.



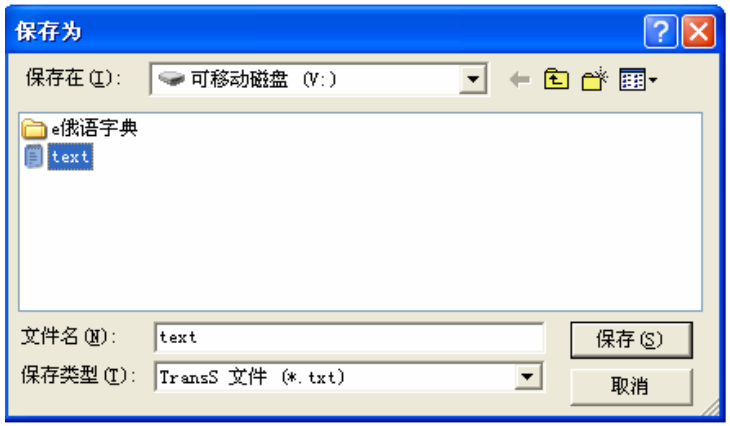
Работа с открытием документа

Для открытия документа используется кнопка  , нажав которую вы получаете окно открытия документа.

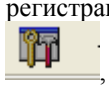


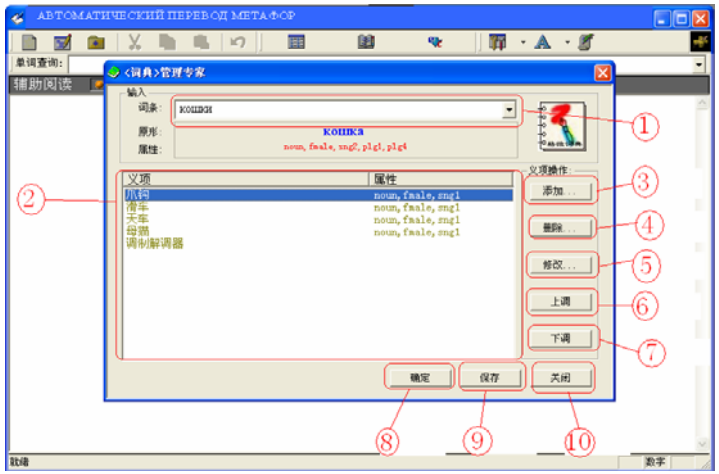
Работа с сохранением документа

Для сохранения документа используется кнопка  , нажав которую вы получаете окно открытия документа.



Работа с регистрацией новых слов и изменения свойств слов

Для регистрации новых слов и изменения свойств слов используется кнопка  , нажав которую вы получаете окно регистрации новых слов и изменения свойств слов, как это показано ниже.



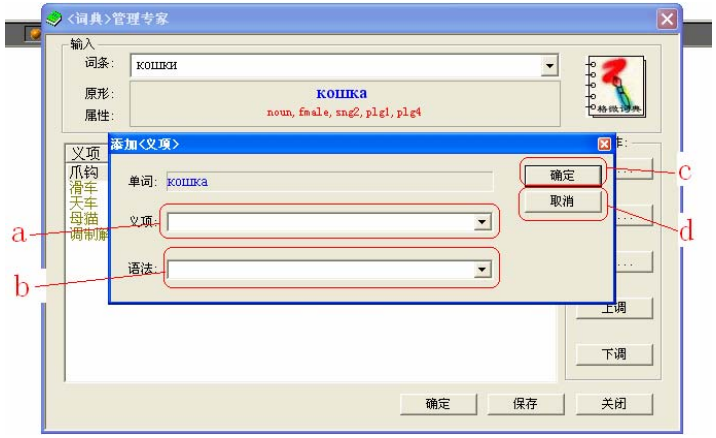
(1) Добавьте новые слова или измените свойства слов. (2) Показать свойства слов. (3) Создать новые свойства слов. (4) Удалить свойства слов. (5) Изменить свойства слов. (6) Изменить место вверх. (7) Изменить место вниз. (8) ОК. (9) Сохранить. (10) Отменить.

Выполнение действий

а) *редактирование сведений о свойствах слов* – нахождение нужной записи в окне показана свойства слов (②), нажатие кнопки «Изменение свойств слов» (⑤), изменение сведений и нажатие кнопки «Сохранить».

(a) Добавьте новые значения слов. (b) Добавьте новые свойства слов. (c) OK. (d) Отмена.

б) *создание новых свойств слов* – нажатие кнопки «Создание новых свойств слов»(□), как это показано на рис, добавление новой записи и нажатие кнопки «Сохранить»(□)



(a) Добавьте новые значения слов. (b) Добавьте новые свойства слов. (c) ОК. (d) Отмена.

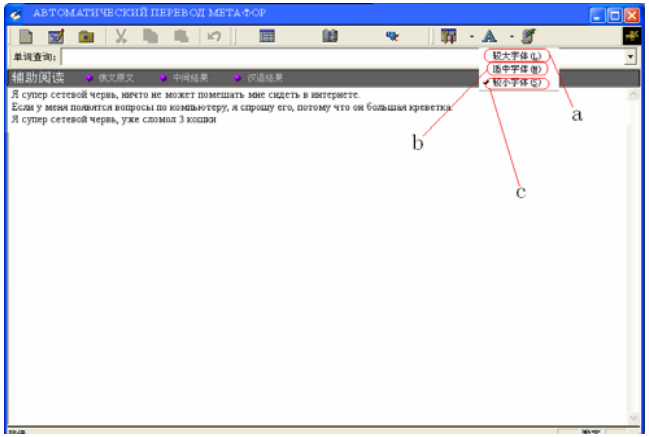
в) *удаление записи о свойствах слов* - нахождение нужной записи в окне показаны свойства слов(□) и нажатие кнопки «Удаление свойств слов»(□), как это показано на рис, и нажатие кнопки «ОК» или «Отмена».

(a) ОК. (b) Отмена.

г) *изменение места о свойствах слов* - нахождение нужной записи в окне показана свойства слов (□), и нажатие кнопки «Изменение место на вверх» (□) или «Изменение место на вниз» (□)

Работа с размером шрифта:

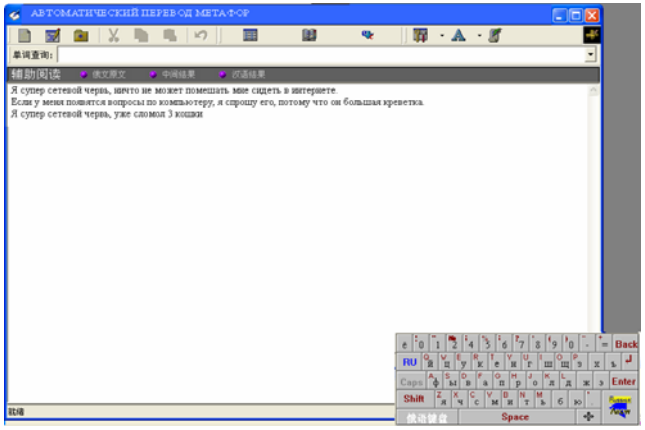
Для изменения размера шрифта используется кнопка  , нажав которую вы получаете три варианта выбора – крупный, средний и мелкий, выбрав нужный можно получить размер шрифта текста в основной колонке.



(a) крупный (b) средний (c) мелкий

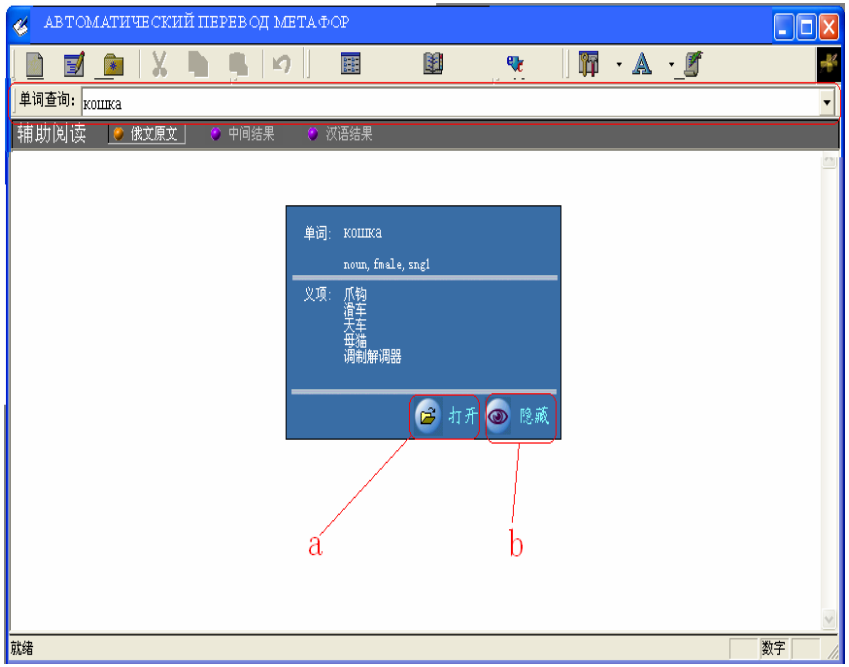
Работа с русской клавиатурой:

Для использования русской клавиатуры используется кнопка  , нажав которую можно получить русскую клавиатуру на экране, чтобы написать русские буквы.



Работа со словарем:

Словарь работает в режиме русского языка, т.е. в окне словаря вы напишете русские метафоры и нажав кнопку «Enter» на клавиатуре появляются значение на китайском.



a) окно регистрации новых слов и изменения свойств слов
b) Отмена

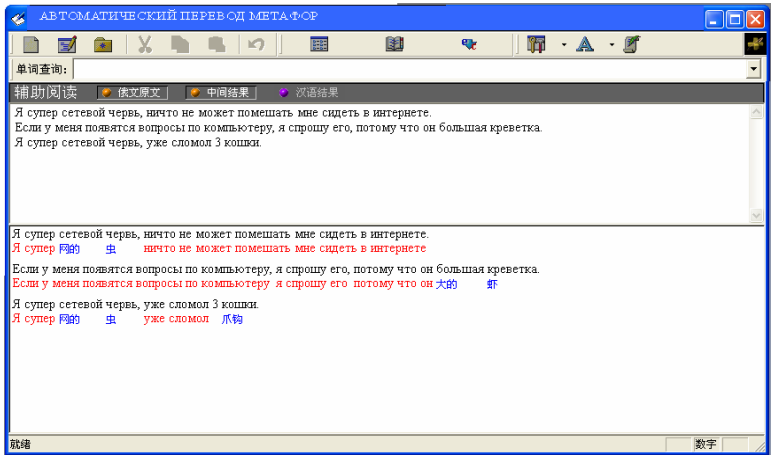
Если вы хотите изменение свойств слов, можете выбирать кнопку «окно регистрации новых слов и изменения свойств слов» (a), и ведет к появлению формы «окно регистрации новых слов и изменения свойств слов».

Работа с автоматическим переводом метафор:

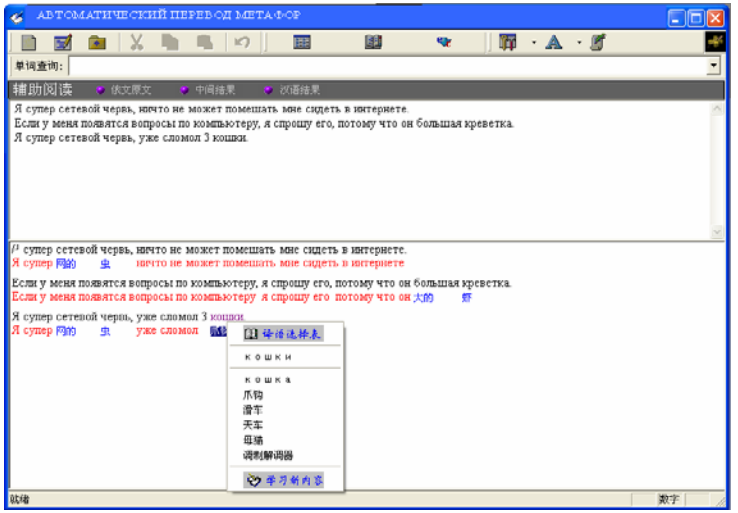
Автоматический перевод метафор работает в режиме русского языка, т.е. используется кнопка 

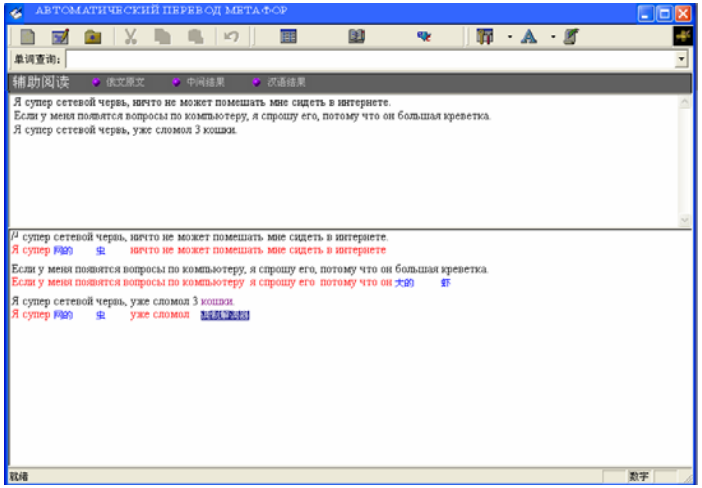
и в окне автоматического перевода метафор вы напишете русские тексты. Потом вы нажимаете кнопку 

и в окне автоматического перевода метафор можно получить переводы метафор на китайском, под русским текстом показываются синие китайские значения.



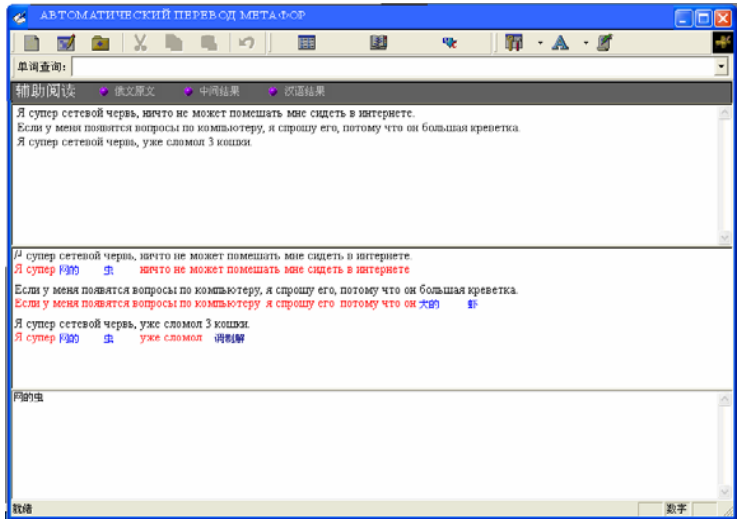
Чтобы изменить китайские значения переводов слов нужно однажды щелкнуть правой кнопкой мыши по интересующей вас метафоре. После одного щелчка в основной форме откроется выбор значений слов, и вы можно выбрать самое точное китайское значение слова под русским текстом.



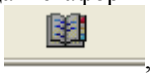


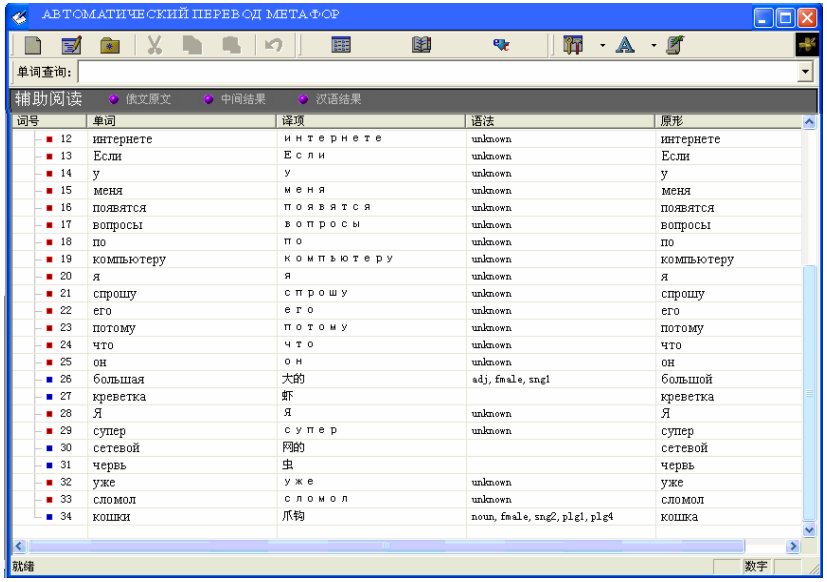
Чтобы получить китайские значения переводов слов нужно нажать в правом верхнем кнопку 

, и один раз щелкнуть левой кнопкой мыши по интересующей вас метафоре. После одного щелчка под русским текстом в основной форме показывается китайское значение слова.



Работа с быстрым словарем:

Когда в окне автоматического перевода метафор вы напишете русские тексты, можете использовать кнопку  , нажав которую можно получить все китайские значения метафор.



Работа с быстрым автоматическим переводом метафор:

Когда в окне автоматического перевода метафор вы напишете русские тексты, можете использовать кнопку  , нажав которую можно получить переводы метафор на китайском, под русским текстом показываются синие китайские значения.

Литература

1. *Ван Чэнь, Гуань Ли.* Китайские метафоры информатики / Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. Выпуск 8 // Сост. и ред. Ю.Н.Филипповича. — М.: «CLAIM», 2006. — С.63–83.

Е.А.Выломова

СИСТЕМА РАСПОЗНАВАНИЯ СЕМИОГРАФИЧЕСКИХ ПЕСНОПЕНИЙ*

Одной из важных и в высшей степени интересных задач в изучении музыкальной культуры древней Руси является раскрытие мелодического содержания рукописных певческих книг XII— XVII веков, в большом количестве сохранившихся до наших дней.

Мелодии в этих книгах записаны посредством безлинейных нотных систем, вырабатывавшихся на Руси на протяжении нескольких столетий. Древние нотации давно уже вышли из употребления и забыты, но с развитием семиотики и музыкальной палеографии они все чаще становятся объектами исследований.

Слово «семиография» означает известный условно принятый способ письма и выражения определенных музыкальных звуков и их взаимоотношений. Фрагмент семиографического песнопения представлен на рис.1. Структурно песнопение состоит из следующих частей:

- строк знамен (семиографических знаков), обозначающих нотные знаки;
- строк текста, которым сопоставляются знамена;
- в структуре знамен можно особо выделить пометы, обозначающие длительность или высотность нот;

Для решения задач автоматизированного ввода данного типа рукописей была разработана система распознавания семиографических песнопений (СРСП), которая позволяет автоматизировать процесса распознавания наиболее простой части песнопений – помет, общее число которых не превышает десяти.

В статье основное внимание уделено описанию и сравнению математических моделей и алгоритмов, легших в основу СРСП. В качестве теоретической базы выбран подход к распознаванию на основе правил классификации, которые формулируются и выводятся в терминах математической статистики.

---

* Научный руководитель к.т.н., доц. А.Ю.Филиппович. Работа выполнена в рамках курсового проекта в 2006–2007 уч.г.

Конкретными методами распознавания были выбраны нейронные сети (НС) трех типов:

- Многослойный персептрон
- НС с общей регрессией
- Вероятностная НС

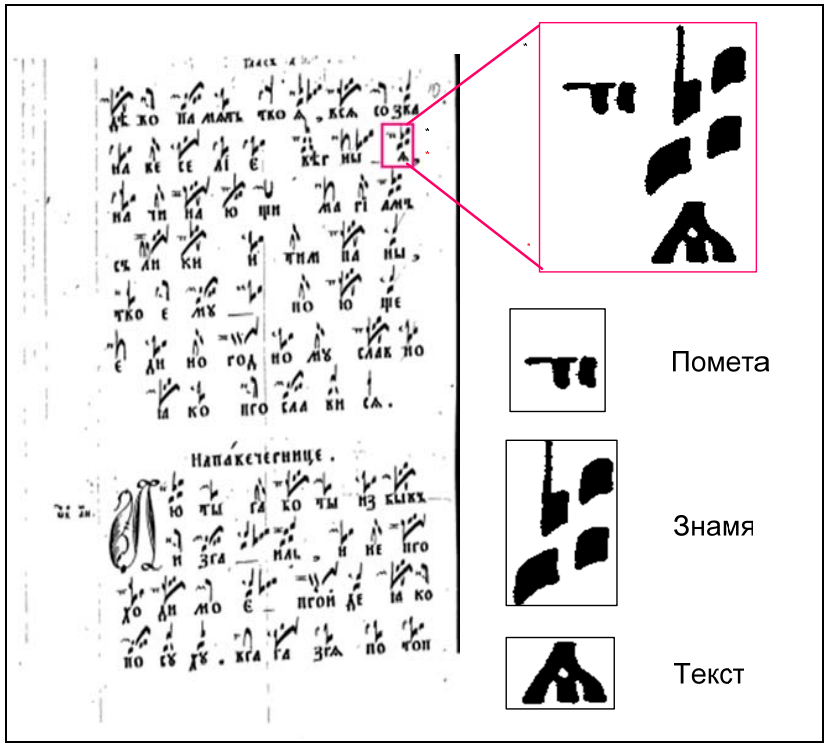


Рис. 1. Структура семиографического песнопения

Многослойный персептрон

В качестве теоретического основания этой модели НС выступают теорема А.Н.Колмогорова, которая гласит, что любую непрерывную функцию на d-мерном кубе можно представить в виде суперпозиции функций двух переменных, и теорема Новикова (применительно к персептрону Розенблатта).

Теорема А.Н.Колмогоров, 1957. Для любой непрерывной функции на d-мерном кубе существует вычисляющая ее трехслойная нейронная сеть с (d+1)(2d+1) скрытыми нейронами.

Теорема Новикова. Пусть задана бесконечная последовательность примеров с элементами, удовлетворяющими неравенству $|u_i| < D$, где U- пространство признаков. Предположим, что существует гиперплоскость с коэффициентами w_0, которая точно разделяет элементы последовательности и удовлетворяет условию $\min\limits_{(y,u)\in\{\hat{Y},\hat{U}\}} \dfrac{y(w_0 * u)}{|w_0|} \geq \rho_0$,

где $\left\{\hat{Y},\hat{U}\right\} = (y_1,u_1),...,(y_l,u_l)$, $\rho_0 > 0$

Затем, используя итеративно процедуру $w(t) = \begin{cases} w(t-1), y_t(w(t-1) * u_t) > 0 \\ w(t-1) + y_t u_t, else \end{cases}$, персептрон строит гиперплоскость, которая корректно разделяет все примеры бесконечной последовательности. Для создания такой гиперплоскости персептрону требуется $M = \left[\dfrac{D^2}{\rho_0^2}\right]$ коррекции.

На рис. 2 представлена типовая схема персептронной сети, реализация которой описывается следующим формализмами:

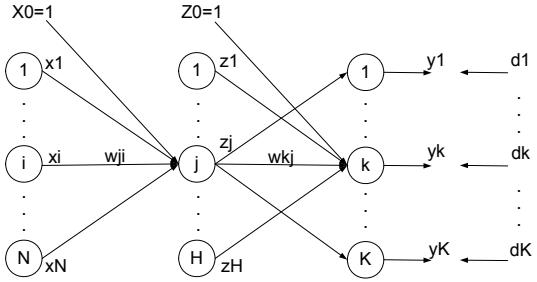


Рис. 2. Структура персептрона

X_i – i-ый нейрон входного слоя, Z_j – j-ый нейрон скрытого слоя, Y_k – k-ый нейрон выходного слоя, D_k- k-ый желаемый выход.

Состояние j-го нейрона скрытого слоя: $$net_j = \sum_{i=0}^{N} w_{ji} x_i$$

Ввыход j-го нейрона скрытого слоя: $$z_j = f_h(net_j)$$

Состояние k-го нейрона выходного слоя: $$net_k = \sum_{j=0}^{H} w_{kj} z_j$$

Выход k-го нейрона выходного слоя: $$y_k = f_0(net_k)$$

Расчет ошибки сети: $$E(w) = \frac{1}{2} \sum_{k=1}^{K} (d_k - y_k)^2$$

Пересчет весов для выходного слоя:

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} = \eta (d_k - y_k) f'_o(net_k) z_j$$

, .

Пересчет весов для скрытого слоя:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \quad \frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial z_j} \frac{\partial z_j}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}}$$

, ,

$$\frac{\partial E}{\partial z_j} = -\sum_{k=1}^{K} (d_k - y_k) \frac{\partial y_k}{\partial z_j} = -\sum_{k=1}^{K} (d_k - y_k) f'_o(net_k) w_{kj}$$

$$\Delta w_{ji} = \eta [\sum_{k=1}^{K} (d_k - y_k) f'_o(net_k) w_{kj}] f'_h(net_j) x_i$$

Обучение и работа сети

Обучение НС осуществляется с помощью алгоритма обратного распространения ошибки, который схематично представлен на рис. 3

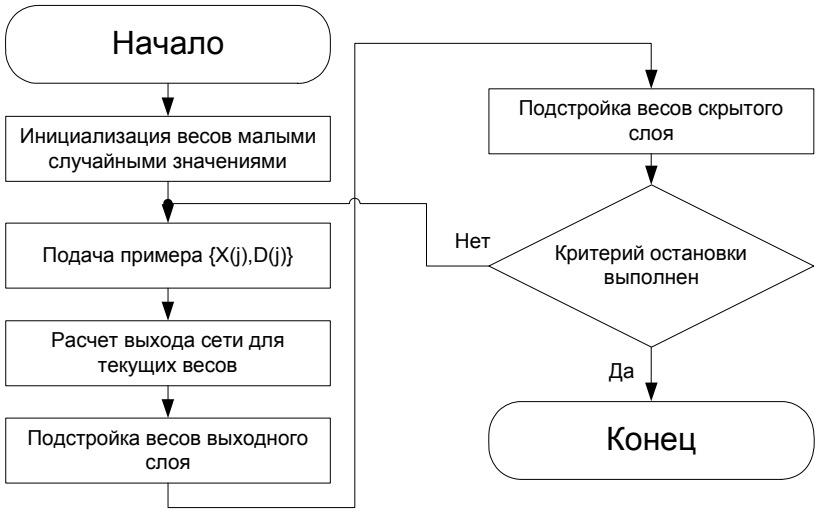


Рис. 3. Алгоритм обратного распространения

Шаг 1. Инициализация весов малыми случайными значениями.

Шаг 2. Задать значения входов $\{x_1,\ldots,x_N\}$ и соответствующих желаемых выходов $\{d_1,\ldots,d_K\}$

Шаг 3. Посчитать ответ сети для текущих значений весов

$$y_k = f_o\left(\sum_{j=0}^{H} w_{kj} f_h\left(\sum_{i=0}^{N} w_{ji} x_i\right)\right)$$

Шаг 4. Построить веса выходного слоя

$\Delta w_{kj}(t) = \eta \delta f'_o(net_k) z_j + \mu \Delta w_{kj}(t-1)$, где $\delta_k = d_k - y_k$

Шаг 5. Подстроить веса скрытого слоя

$\Delta w_{ji}(t) = \eta \delta f'_h(net_j) x_i + \mu \Delta w_{ji}(t-1)$, где

$$\delta_j = \sum_{k=1}^{K} (d_k - y_k) f'_o(net_k) w_{kj}$$

Шаг 6. Если критерий остановки не выполнен, то Шаг 2, иначе СТОП.

Нейронная сеть с общей регрессией

НС с общей регрессией была впервые представлена Надараем и Ватсоном. Регрессионная функция, выполненная на независимой переменной X, вычисляет наиболее вероятное значение зависимой переменной Y, основанной на конечном множестве наблюдений X и соответствующих значений Y.

Пусть $f(x,y)$ совместная непрерывная функция плотности вероятности случайного вектора x и скалярного случайного значения переменной y. Пусть X - частное значение случайной переменной x. Тогда регрессия y данного X будет:

$$E[y/X] = \frac{\int_{-\infty}^{+\infty} y * f(X,y)dy}{\int_{-\infty}^{+\infty} f(X,y)dy}$$

Обычно совместная плотность $f(x,y)$ неизвестна. Тогда оценочная функция будет иметь вид:

$f_n(x) = \frac{1}{n\lambda}\sum_{i=1}^{n}\varphi(\frac{x-x_i}{\sigma})$, где x_i - независимые одинаково распределенные случайные переменные с абсолютно непрерывной функцией распределения. Весовая функция φ должна быть ограниченной и удовлетворять следующим условиям:

$$\int_{-\infty}^{+\infty}|\varphi(y)|dy < \infty \; ; \; \lim_{y\to\infty}|y\varphi(y)| = 0 \; ; \; \int_{-\infty}^{+\infty}\varphi(y)dy = 1$$

Функция $\sigma = \sigma(n)$ должна быть выбрана таким образом, чтобы

$$\lim_{n\to\infty}\sigma(n) = 0 \; ; \; \lim_{n\to\infty} n\sigma^2(n) = \infty$$

Одной полезной формой весовой функции φ является функция плотности ядра (Гауссиан). Парзен показал, что эти оценочные функции состоятельны.

$$\hat{f}(X,Y) = \frac{1}{(2\pi)^{(p+1)/2}\sigma^{p+1}} * \frac{1}{n}\sum_{i=1}^{n}\exp\left[-\frac{(X-X^i)^T(X-X^i)}{2\sigma^2}\right] * \exp\left[-\frac{(Y-Y^i)^2}{2\sigma^2}\right]$$

, где p является размерностью вектора x, а n количеством наблюдений.

Определим скалярную функцию D_i^2

$$D_i^2 = (X-X^i)^T(X-X^i)$$

и , подставив ее, получим

$$\hat{Y}(X) = \frac{\sum_{i=1}^{N} Y^i \exp(-\frac{D_i^2}{2\sigma^2})}{\sum_{i=1}^{N}\exp(-\frac{D_i^2}{2\sigma^2})}$$

Данная регрессия носит название оценочной функции Надарая-Ватсона. Оценка $\hat{Y}(X)$ может быть представлена как взвешенное среднее всех наблюдаемых значений Y^i.

Первая реализация сети впервые была представлена Спехтом и описывается ниже:

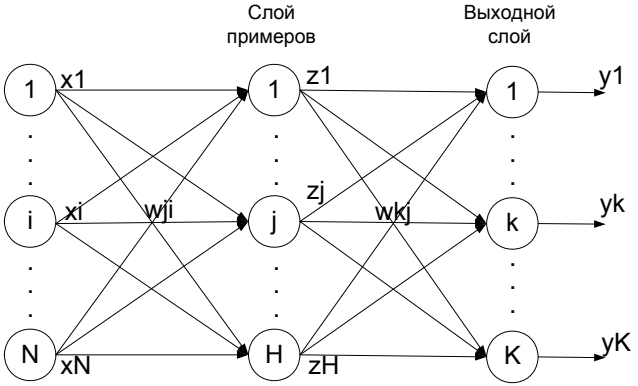


Рис. 4. Структура нейронной сети с общей регрессией

$\mathbf{X_i}$ – i-ый нейрон входного слоя, $\mathbf{Z_j}$ – j-ый нейрон слоя примеров, $\mathbf{Y_k}$ – k-ый нейрон выходного слоя.

Выход j-го нейрона слоя примеров:

$$z_j = \exp\left(-\frac{\sum_{i=1}^{N}(w_{ji} - x_i)^2}{2\sigma^2}\right)$$

Выход k-го нейрона выходного слоя:

$$y_k = \frac{\sum_{j=1}^{H} w_{kj} z_j}{\sum_{j=1}^{H} z_j}$$

Вероятностная нейронная сеть

Вероятностная нейронная сеть, часто называемая байесовской, основана на аналогичных теоретических принципах, что и сеть с общей регрессией, т.к. в ней используются оценочные функции распределения вероятности Парзена. Правдоподобие неизвестного вектора, принадлежащего данному классу, может быть выражено как

$$f_i(x) = \frac{1}{(2\pi)^{p/2}\sigma^p} * \frac{1}{n_i}\sum_{i=1}^{n_i}\exp\left[-\frac{(x - x_{ij})^T(x - x_{ij})}{2\sigma^2}\right],$$ где i – номер класса, j – номер паттерна, n_i - количество обучающих векторов в i – классе, x - тестовый вектор.

Ниже представлена реализация сети

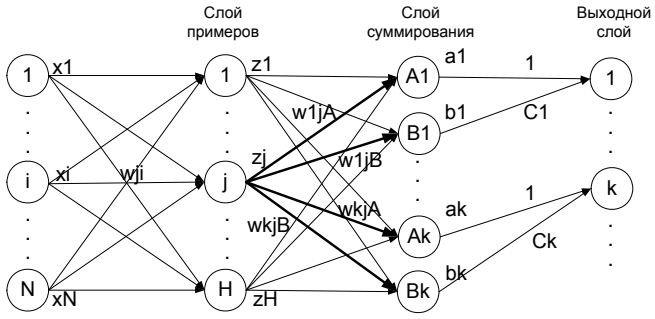


Рис. 5. Структура вероятностной нейронной сети

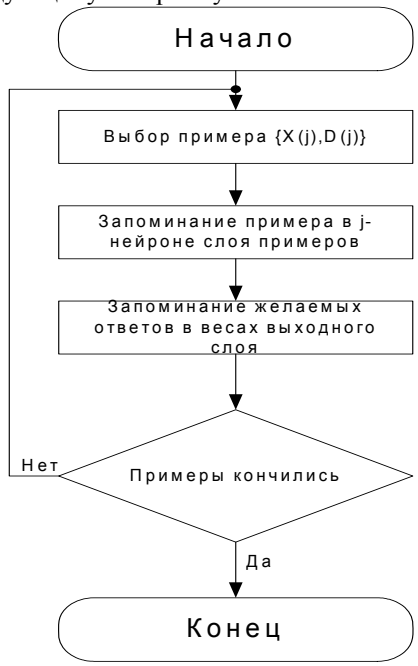
$\mathbf{X_i}$ – i-ый нейрон входного слоя, $\mathbf{Z_j}$ – j-ый нейрон слоя примеров, $\mathbf{A_k}$ – k-ый нейрон слоя суммирования, принадлежащий данному классу, $\mathbf{B_k}$ – k-ый нейрон слоя суммирования, не принадлежащий данному классу, $\mathbf{Y_k}$ – k-ый нейрон выходного слоя.

Выход j-го нейрона слоя примеров: $$z_j = \exp\left(-\frac{\sum_{i=1}^{N}(w_{ji} - x_i)^2}{2\sigma^2}\right),$$

Для всех нейронов слоя суммирования: $$a_k = \sum_{j=1}^{H} w_{kj}{}^A z_j \qquad b_k = \sum_{j=1}^{H} w_{kj}{}^B z_j,$$

Для всех нейронов выходного слоя: $$y_k = sign(a_k + C_k b_k), \qquad C_k = -\frac{l_{Bk}}{l_{Ak}}\frac{h_{Bk}}{h_{Ak}}\frac{n_{Ak}}{n_{Bk}}$$

Обучение вероятностных (байесовых) сетей осуществляется по следующему алгоритму:



Шаг 1. Выбрать j-ый пример {X(j),D(j)}

Шаг 2. Запомнить пример в весах j-го нейрона в слое примеров $w_{ji}=x_i(j)$

Шаг 3. Запомнить желаемые ответы в весах выходного слоя $w_{kj}=d_k(j)$

Шаг 4. Если остались еще примеры, то Шаг2, иначе СТОП.

Результаты обучения и сравнение нейронных сетей

На базе анализа литературы были выявлены достоинства и недостатки сетей, которые представлены в таблице ниже.

	Рыхлые данные	Малое количество данных	Локальные минимумы	Большое число входов
Персептрон	-	+	+	+
НС с общей регрессией	+	+	–	–
Вероятностная НС	+	+	–	–

В процессе разработки системы был проведен ряд сравнительных запусков системы при различных параметрах нейросетей и признаков.

Для персептрона скорость и момент обучения были выбраны на основе следующих данных. Полученных в результате экспериментов:

- Зависимость количества неверных ответов от скорости обучения (момент=0.9)
- Зависимость количества неверных ответов от момента обучения (скорость =0.01)

При этом следует отметить, что при одновременном изменении этих параметров в допустимых рамках, можно улучшить результат.

Например, при скорости обучения 0.1 и моменте 0.5 количество неверных ответов сети будет равно 0.1.

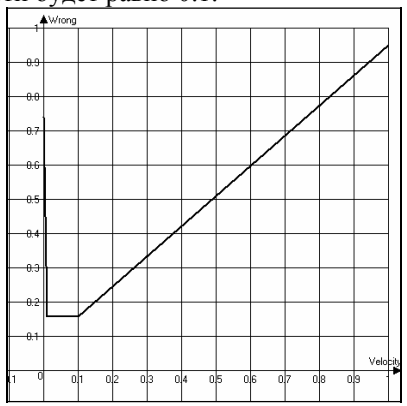


Рис. 6. Зависимость процента распознавания от скорости при постоянном моменте

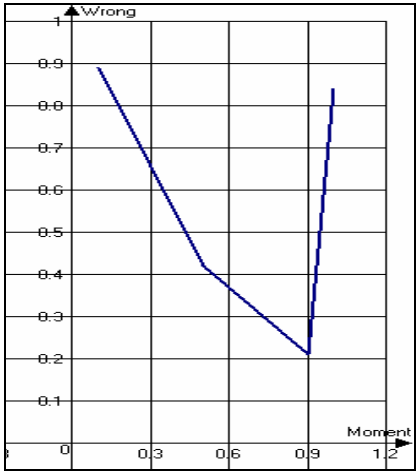


Рис. 6. Зависимость процента распознавания от момента при постоянной скорости

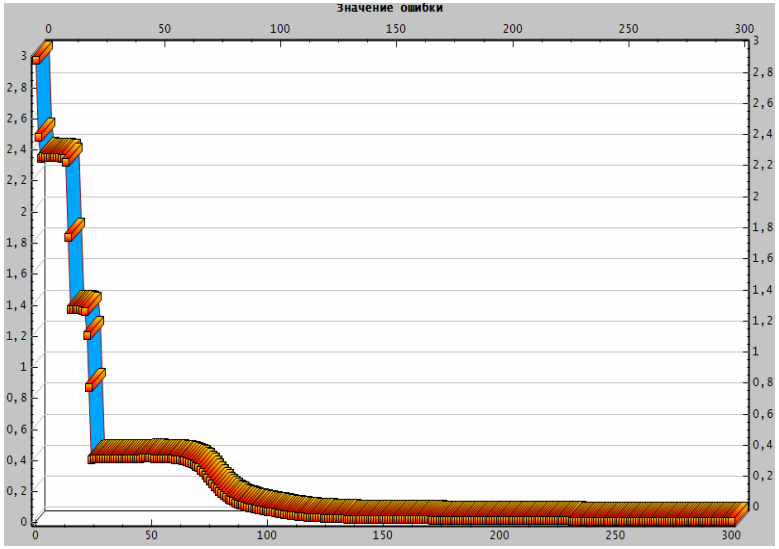


Рис. 7. Персептрон. Уменьшение ошибки в процессе обучения

Ниже приведены результаты работы сетей при количестве признаков=25

Вероятностная НС(1 эпоха)

Слой	Входной слой	Скрытый слой	Выходной слой
Нейронов	25	110	7
Символ	Среднее значение выхода	Максимальное значение выхода	Минимальное значение выхода
С	77.77800735	99.99999987	33.33402255
М	99.99999994	100	99.99999981
Н	99.99999999	100	99.99999995
П	79.16640769	99.99999961	50.00310248
Р	99.99922433	100	99.99689771
В	99.99999884	99.99999884	99.99999884

Статистика распознавания: правильно – 17; неправильно – 2.

Персептрон (300 эпох, скорость=0.01, момент=0.9)

Слой	Входной слой	Скрытый слой	Выходной слой
Нейронов	26	14	7
Функция	Экспоненциальная		
С	0.631956791	0.640731412	0.623182169
М	0.946009168	0.955919016	0.93682458
Н	0.768222496	0.940875576	0.14833814
П	0.777250872	0.900827998	0.469283307
Р	0.843146703	0.947454024	0.622591368
В	0.80032522	0.80032522	0.80032522

Статистика распознавания: правильно – 16; неправильно – 3.

НС с общей регрессией(1 эпоха)

Слой	Входной слой	Скрытый слой	Выходной слой
Нейронов	25	110	7
Символ	Среднее значение выхода	Максимальное значение выхода	Минимальное значение выхода
С	0.777780073	0.999999999	0.333340226
М	0.999999999	1	0.999999998

Н	1	1	1
П	0.791664077	0.999999996	0.500031025
Р	0.999992243	1	0.999968977
В	0.999999988	0.999999988	0.999999988

Статистика распознавания: правильно – 17; неправильно – 2.

В результате было установлено, что при решении данной задачи лучше работают вероятностная нейронная сеть и сеть с общей регрессией. Следует отметить, что особенностью данной задачи являлось ограниченное и весьма небольшое количество примеров в обучающей выборке.

Также было замечено, что количество признаков не должно быть слишком большим или слишком малым, т.к. при этом классификация существенно затрудняется.

И.В.Даньшина, М.В.Даньшина

СТАТИСТИЧЕСКОЕ ИССЛЕДОВАНИЕ ЗНАМЕННОЙ НОТАЦИИ*

Введение

Современная запись мелодий производится с помощью нот, располагающихся на специальных линейках, что обеспечивает удобство чтения и обучения нотной грамоте: любая нота взаимно однозначно определяет высотность, существуют специальные обозначения для ритма и скорости исполнения.

Однако к такой форме записи пришли не сразу. Первоначально существовала греческая система описания музыки на основе букв, которые затем заменили на специальные графические обозначения невмы. При этом каждая невма служила лишь средством напоминания певцу уже известной ему мелодии и наглядным указанием на ее восходящее или нисходящее направление.

В русской церкви невмы употреблялись в виде крюковой или знаменной нотации. В 17 веке, в целях удобства, были введены так специальные пометы для указания высоты исполнения знамени, а уже в 18 веке стали постепенно отказываться от знаменной формы записи, переходя к привычным на сегодняшний день пяти линейкам.

Вместе с тем сохранившиеся до наших дней древнерусские музыкальные рукописи в знаменной нотации до сих пор полностью не систематизированы и не расшифрованы. Решение этой проблемы – довольно сложная и трудоемкая задача, но в то же время важная и интересная.

Обзор материалов в Интернет подтверждает, что данным вопросом занимается достаточно большое количество людей, некоторые из которых разработали свои шрифты и постепенно переводят в электронный вид знаменные песнопения, занимаются разработкой методик обучения чтению двухзнаменной (знаменной и нотолинейно) нотации.

Цель нашей работы – изучение древних рукописей в знаменной нотации с использованием статистических методов исследований для определения синтаксиса и семантики песнопений.

---

* Научный руководитель к.т.н., доц. А.Ю.Филиппович. Работа выполнена в 2006–07 уч.г.

В качестве исходных данных исследования взяты пометные песнопения первые два гласа первой части Круга церковного древнего знаменного пения в шести частях (под редакцией Д.В. Разумовского) [2].

Исследование семантики песнопений

Для получения статистических результатов была использована специальная компьютерная программа AndrewTools [1] (Рис. 1 и Рис. 2). С помощью этого лингвистического редактора над текстом можно выполнять специальные операции: создавать словники, конкордансы, сортировать текст по специальному (собственному) алфавиту, создавать частотные словники и другие. К сожалению, из-за специфичности предметной области, нельзя обойтись использованием только этой программы - нужно также преобразовывать промежуточные результаты в таких программах как MS Word, MS Excel и Блокнот.

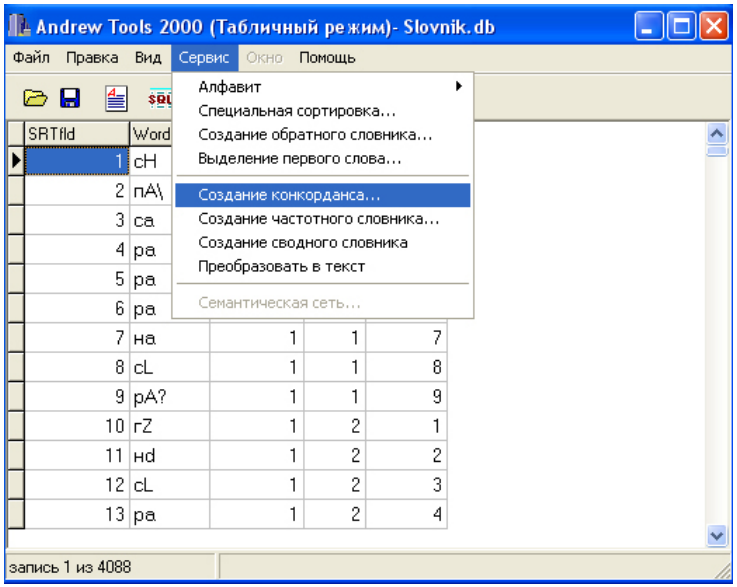


Рис. 1. Программа Andrew Tools 2000. Табличный режим

В результате проделанной работы были получены статистические данные, которые отражают частоту соответствий знамен песнопений и отдельных слогов, а также входящих в них гласных букв. Фрагменты результирующих таблиц представлены на рис. 3 и 4.

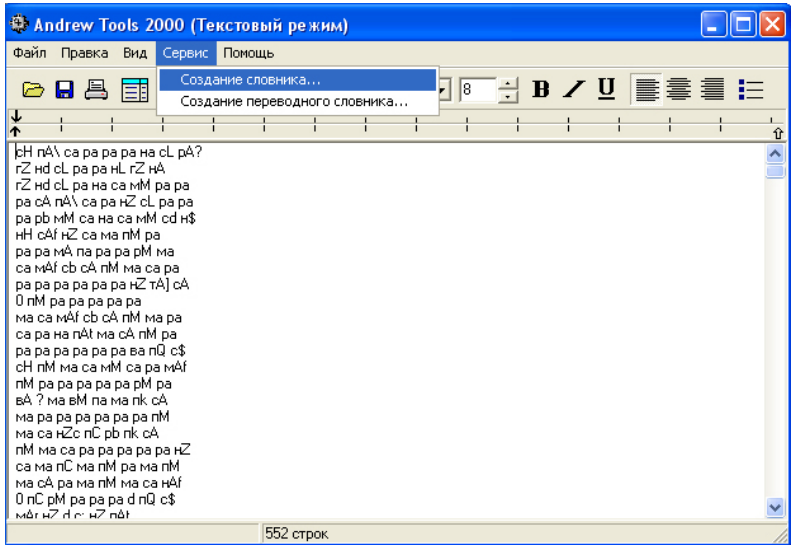


Рис. 2. Программа Andrew Tools 2000. Текстовый режим

	Глас 1			Глас 2			Гласы 1 и 2		
1									
2									
3		—	11		—	20		—	20
4			9			8			14
5			8			7		—	17
6		—	6			7			11
7			6		—	6			10
8			6		—	6			9
9			6			6			9
10			6		—	6			8
11		—	5			5			8
12			5		—	5			8
13			5			5			7
14			5			5			6
15		—	4			5			6
16			4			5			6

Рис. 3. Частота встречаемости пар слог-знамя

	Глас 1		
1			
2			
3		о	38
4		о	38
5		а	32
6		и	27
7		е	27
8		е	26
9		е	21
10		а	18
11		о	16
12		и	16
13		и	15
14		е	15
15		о	14
16		о	14

Рис 4. Частота встречаемости пар гласная-знамя

Кроме того, были получены данные о частоте встречаемости слогов и их комбинаций (конкордансов)[5].

	гласная	слог	частота	гласная	слог	частота	гласная	слог	частота	гласная	слог	частота
2	о		333	е		323	и		291	а		236
3		го	34		е	49		и	71		на	32
4		по	25		не	23		ди	26		ва	14
5		со	21		ие	22		ти	16		ма	14
6		во	19		те	19		ми	15		да	14
7		но	18		ге	12		ли	15		ра	12
8		ко	18		ве	12		ни	14		ела	8
9		бо	14		пре	10		си	12		ша	8
10		ро	12		че	8		нз	12		ета	7
11		го	12		ке	7		при	8		ела	7
12		ито	9		емз	6		ети	7		га	5
13		мо	8		кре	6		ши	6		та	5
14		кот	8		де	6		ыи	6		ка	5
15		коз	7		ке	5		ки	6		а	4
16		ело	6		це	5		шимз	4		ла	4
17		до	6		зем	5		чи	4		за	4

Рис. 5. Оценка частоты встречаемости слогов (Глас 1)

[5] Из-за ограничений инструментов исследования комбинации слогов не всегда образуют слова, что требует проведение ручной корректировки.

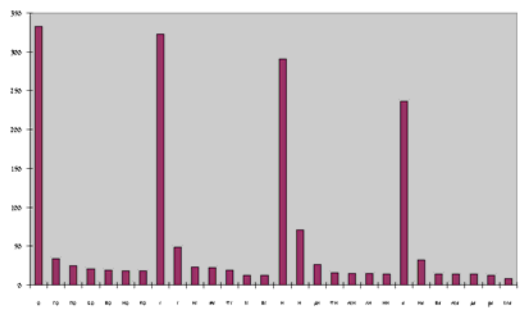


Рис. 6. Диаграмма встречаемости слогов (Глас 1)

Цель проведенного анализа – выявить наличие или отсутствие закономерности между слогами и знаменами, а также между гласными буквами слогов и знаменами.

Из рисунка 3 видно, что максимальное число повторений по двум гласам 20. Если сравнить это число с общим количеством элементов в таблице (4083), то можно сделать вывод, что частотного соответствия между знаменами и слогами нет.

Теперь рассмотрим соответствие гласной знамени. Буква **О** встретилась в первом гласе 333 раза. Из них с 38 раз и с тоже 38 раз. Учитывая, что и – наиболее часто встречаемые крюки, то по результатам анализа (см. рисунок 4), был сделан вывод, что скорее всего однозначного или устойчивого соответствия между гласными и отдельными знаменами нет.

нї	17
яко	9
Госпо	8
нні	7
роди	7
вокру	6
тебѣ	6
сан	6
споди	6
нпо	6
нго	5
шй	5
сне	5

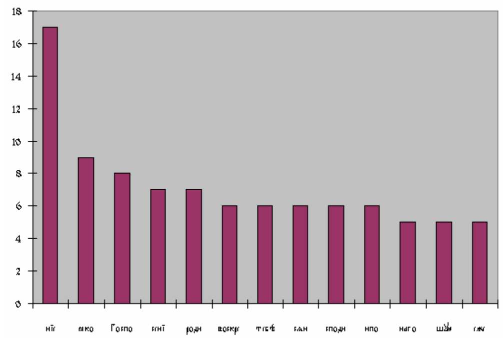


Рис. 7. Оценка частоты встречаемости комбинации слогов. Конкорданс из двух слогов.

[illegible]	4
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2
[illegible]	2

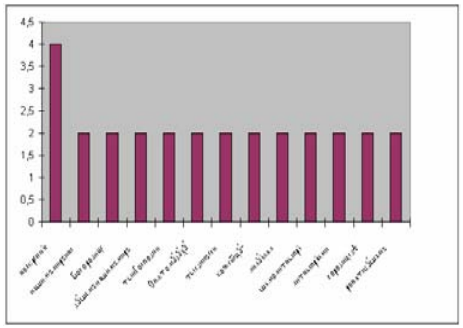


Рис. 8. Оценка частоты встречаемости комбинации слогов. Конкорданс из пяти слогов.

Гипотеза о том, что некоторые слоги поются всегда одинаково, не подтвердилась, но это не исключает того, что некоторые слова исполняются одними и теми же знаменами. Для проведения этого анализа необходима программа, которая бы точно определяла, является ли полученный конкорданс словом, или это просто набор слогов. Когда проблема выделения правильных слов будет решена, то, сравнив конкордансы с соответствующими им крюками, мы сможем ответить на поставленный вопрос. Пока такой возможности нет, мы можем только посмотреть частоту встречаемости комбинаций слогов.

Статистика по знаменам

Всего в первом и втором гласах встречается 8439 элемента, из них 4759 знамени, а остальные 3680 элемента – это различные пометы. Каждому слогу может соответствовать как одно знамя, так и комбинация из нескольких знамен. Всего различных комбинаций встретилось 4089, причем у большей части (3400) присутствуют пометы, а у 689 – нет.

Кроме помет, у знамени может быть признак, который указывает, к какой высотной области он относится. Всего существует четыре певческих области: «простая», «мрачная», «светлая» и «тресветлая», каждая из которых разделена на три подобласти и аналогично они называются «простая», «светлая» и «тресветлая». Существуют специальные указания, как должен выглядеть признак для каждого знамени, если он существует, а если признак не указан, то знамя употребляется в простой области. Итак, всего знамен с признаками было обнаружено 94. Из них «светлых» – 41, тресветлых – 53.

По итогам исследования количество различных знамен составляет 91, а различных помет 9: одна исполнительная и восемь высотных. Всего в первом и втором гласах 4759 знамен, из которых уникальными являются 4089.

На рисунке 10 отображена гистограмма, на которой представлены 30 наиболее часто встречающихся крюков:

	651		102		50
	595		98		46
	387		91		44
	387		90		40
	283		77		38
	270		68		37
	242		60		32
	197		58		30
	154		57		28
	111		55		27

Рис. 9. Наиболее часто встречающиеся знамена

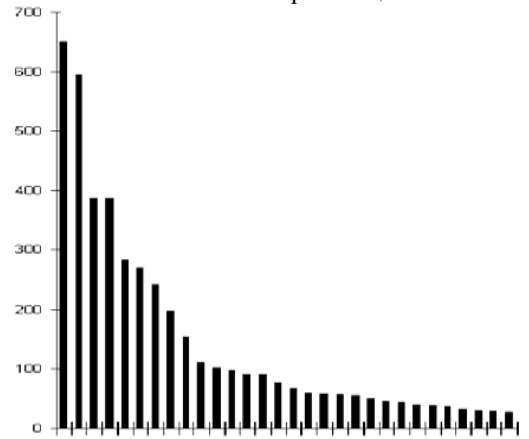


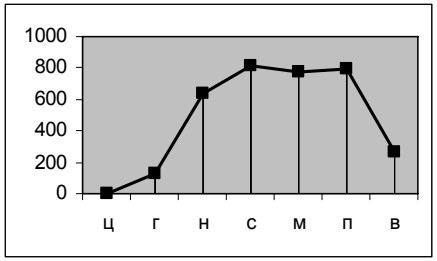
Рис. 10. Гистограмма наиболее часто встречающихся знамен

Высотные пометы 1 и 2 гласов

Всего в первом и втором гласах встречается 3407 высотных помет. В работах Смолякова Б.Г. [3] высказывается предположение, что в каждом гласе доминирует определенная высота исполнения. Проверим это,

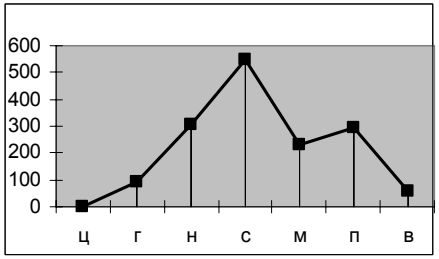
построив гистограммы для первого и второго гласов по отдельности, а также их совместную гистограмму.

Для обоих гласов:



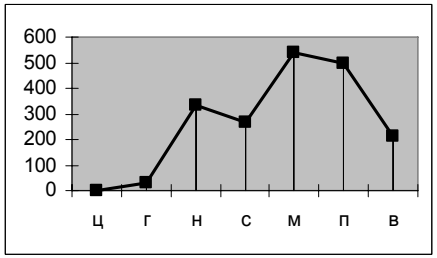
ц	2	си
г	123	до
н	634	ре
с	812	ми
м	773	фа
п	794	соль
в	269	ля

Для первого гласа:



ц	0	си
г	94	до
н	303	ре
с	546	ми
м	233	фа
п	296	соль
в	55	ля

Для второго гласа:



ц	2	си
г	29	до
н	331	ре
с	266	ми
м	540	фа
п	498	соль
в	214	ля

Рис. 11. Гистограммы встречаемости высотных помет

Эти гистограммы подтверждают гипотезу о том, что некоторые ноты встречаются чаще других, а именно: в первом гласе преобладают звуки «ми», «ре» и «соль», а во втором – «фа», «соль» и «ре». Если посмотреть на общую гистограмму, то в середине скачкообразный вид гистограммы выравнивается, и становится ясно видно, что 4 ноты из семи

практически с одной и той же высокой частотой определяют высоту исполнения первых двух гласов.

Конкордансы

Следующим этапом изучения знаменной семиографии было выявление попевок, которыми в древнерусском певческом искусстве называются мелодико-ритмические обороты, устойчиво повторяющиеся в различных песнопениях и фиксируемые неизменными, только им присущим начертанием. Попевка является определяющим конструктивным элементом гласа.

Одна из особенностей знамен, содержащихся в попевке – это то, что знамя может потерять свои характерные черты исполнения, которыми оно обладало при одиночном употреблении.

Каждому гласу характерен свой уникальный набор попевок, хотя некоторые из них могут соответствовать нескольким гласам. Каждая попевка состоит из двух частей: первая часть называется «доступка», а вторая – «основа». На базе неизменной «основы» может быть построено несколько попевок, у которых будет варьироваться первая часть [4].

Выявление попевок – очень трудная задача, которая требует детальной проработки. В ходе данной работы были проделаны следующие этапы по ее решению:

1. Поиск литературы, которая содержит списки известных попевок, к сожалению, не дал значительных результатов, т.к. приблизительно из 100 просмотренных попевок встретились в 1-2 гласах только 3.

2. На втором этапе был придуман метод поиска сочетаний из нескольких знамен, которые могли бы быть попевками. Для этого были составлены конкордансы из 2, 3,…,10 знамен.

Было определено, что максимальное количество знамен в конкордансе должно быть 9, т.к. при 10 не встречается ни одной комбинации, повторяющейся более одного раза.

Пример поиска попевок. Сначала рассматриваем конкордансы по 8, исключая те, которые состоят только из одного повторяющегося знамени. Выяснилось, что все они имеют общий костяк, состоящий из четырех знамен. Тогда мы проанализировали конкордансы по 5 с этой комбинацией и выбрали из них те, которые встречаются наиболее часто и в разных частях гласов (т.е. в песнопении расположены не рядом). Это делается из тех соображений, что рядом стоящие повторяющиеся комбинации могут и не быть попевкой, являясь простым повторением какой-либо мелодии.

Далее аналогично были рассмотрены все конкордансы длиною в пять знамен, которые встречаются не менее трех раз. В результате проделанной работы удалось выделить следующие попевки:

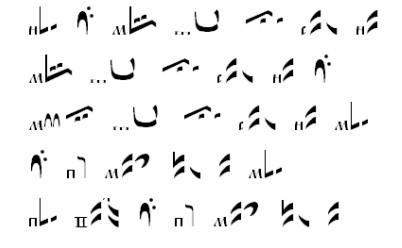


Рис. 12. Выделенные попевки

Заключение

Результаты, полученные в процессе работы, могут быть полезны для решения проблемы расшифровки беспометных записей нот. Несмотря на то, что не было выявлено устойчивой зависимости между слогом (гласной буквой) и знаменем, – это ответ на поставленный вопрос, решение которого до этого не было известно.

Важным результатом является полученная статистика встречаемости слогов, знамен и их комбинаций. Для повышения качества результатов в дальнейшем планируется усовершенствовать шрифты, добавив некоторые знамена и перерисовав имеющиеся.

Материал, с которым пришлось работать, не был единообразно оформлен, поэтому в дальнейшем предполагается сформировать четкую и строгую, но в тоже время понятную методику оформления исходных данных. Это также поможет в создании электронной версии «Круга церковного древнего знаменного пения…».

Литература

1. Филиппович А.Ю. Лингвистический редактор Andrew Tools 2000. Scripta linguisticae applicatae. Проблемы прикладной лингвистики - 2001. Сборник статей. — М.: "Азбуковник", 2001. с. 305-310.

2. Круг церковного древнего знаменного пения в шести частях (под редакцией Д.В. Разумовского) - Общество любителей древней письменности, 1884-1885 гг.

3. Смоляков Б.Г. К проблеме расшифровки знаменной нотации. Вопросы теории музыки. Выпуск 3, М. 1975.

4. http://dyak-oko.mrezha.ru/penie.php

С.А. Иванов

КОМПЕТЕНТНОСТНЫЙ ПОДХОД К СОЗДАНИЮ СИСТЕМ УПРАВЛЕНИЯ ЗНАНИЯМИ В ОБЛАСТИ ИКТ

Информационно-коммуникационные технологии (ИКТ) имеют значительное влияние в целом на экономику за счет увеличения эффективности бизнеса. ИКТ автоматизируют рутинные операции и добавляют «интеллектуальность» во многие современные продукты. В дальнейшем влияние ИКТ на автоматизацию и интеллектуализацию продуктов будет расти. Зачастую внимание руководителей IT-подразделений уделяется исключительно технической стороне вопроса, а необходимость изменений в развитии человеческих ресурсов не всегда принимается в расчет. Вместе с ростом значительности технологического инструментария в рамках почти всех отраслей промышленности, продуктивность и конкурентоспособность компаний, в том числе и крупных международных корпораций, будет все больше зависеть от эффективности использования технологий техническими специалистами, менеджерами и прочими сотрудниками. Поэтому, вопросам компетентности сотрудников необходимо уделять больше внимания, чем это делается сейчас.

Для автоматизации задач подбора и развития персонала используются *системы управления компетенциями*, которые часто входят как компоненты в системы управления знаниями. Подобные системы могут решать различные задачи для нескольких групп людей:

для работодателей:

- подбирать нужных специалистов;
- определять каких знаний и умений не хватает персоналу, с целью их дальнейшего обучения;
- проводить долгосрочное планирование подбора специалистов;
- формировать уровень зарплаты на основании необходимых знаний и навыков специалистов;

для соискателей работы:

- находить работу специалистам в соответствии с их знаниями и умениями;
- выявлять недостающие знания и умения и находить курсы, где их можно получить;
- планировать карьеру и видеть ее перспективы;

для учащихся:

- строить образовательный процесс как обучение компетенциям, которые реально востребованы на рынке труда;

для обучающих:

- своевременно обновлять курсы обучения в соответствии с потребностями на рынке труда.

Преимуществом систем данного класса является возможность формализации знаний на основе компетенций, что позволяет создать единую базу данных, а в перспективе — базу знаний по компетенциям, и использовать ее как центр интегрированной системы управления человеческими ресурсами. Стандартизация описания компетенций позволит автоматизировано обновлять информацию о них, что заметно снизит затраты труда и времени на поддержание перманентной актуальности информации.

Программная реализация единой базы знаний позволяет создавать ее различные представления для разных групп пользователей — работодателей, соискателей, учащихся и обучающих. Для каждого представления могут стоять свои задачи поиска, реализованы различные уровни доступа к редактированию информации и т. п. Данный центр может использоваться как справочник для HR-менеджеров, как помощник специалистам для выявления необходимых знаний для работы в той или иной области или как помощник учащемуся в выборе направления обучения и построении карьеры.

Область ИКТ стремительно развивается, поэтому необходимо постоянно расширять свои знания и совершенствовать навыки. Система управления знаниями позволяет реализовать концепцию обучения в течение всей жизни (lifetime education), помогая определить направление дальнейшего развития и подобрать необходимые курсы, благодаря чему специалисты могут постоянно обладать знаниями и умениями (компетенциями), востребованными в данный момент.

Модели компетенций

Работа по созданию моделей (систем) компетенций ведется в ряде стран. В 1999 г. в рамках Болонского процесса Европейская Комиссия одобрила инициативу Career Space по созданию рабочей группы для

того, чтобы определить новую матрицу специальностей в области ИКТ, соответствующих ей описаний каждой специальности и содержания учебных программ и методов аттестации специалистов. Европейская комиссия в сентябре 2001 г. утвердила ICT Skills Monitoring Group, состоящую из представителей всех стран-членов ЕЭС. Группа проделала работу, основными результатами которой явились:

- выработка общей стратегии определения ИКТ-специальностей;
- анализ существующих специальностей (проекты Socrates и Leonardo da Vinci) и действующих национальных систем ИКТ-специальностей;
- анализ рынка труда ИКТ-специалистов.

Аналогичная работа по созданию концепции системы подготовки специалистов в области ИКТ проводится и на уровне отдельных стран в рамках национальных проектов, например:

- The Skills Framework for the Information Age (SFIA), разработанная при поддержке правительства Великобритании;
- The Advanced IT Training System (AITTS), разработанная в рамках партнерства в отрасли ИКТ при поддержке и под руководством Федерального министерства исследований и образования в Германии;
- The Generic ICT Skills Profiles, разработанные консорциумом Career Space под эгидой Computing Technology Industry Association (CompTIA);
- IT Skills Standards, разработанные National Workforce Centre for Emerging Technologies (NWCET) в США для National Skills Standards Board (NSSB).

Согласно рекомендациям «European e-skills Conference» 2004 года при разработке указанных моделей компетенций необходимо решить следующие проблемы:

- Недостаток (Shortage) — недостаток кадров с необходимым уровнем знаний и умений;
- Разрыв (Gap) — разница между нужным и существующим уровнями кадров;
- Несовпадение (Mismatch) — несовпадение между знаниями и умениями выпускников учебных заведений и требуемыми знаниями и умениями кадров.

При этом для крупных международных компаний стоит и другая проблема — разнородность уровня подготовки специалистов в разных странах.

Задачу обеспечения одинакового уровня подготовки специалистов с гарантированным уровнем знаний в разных странах было призвано решить неформальное образование, в частности, глобальная сертификация, которая бывает двух видов — производителезависимая (vendor related) и производителенейтральная (vendor neutral).

Первый вид сертификатов подтверждает знания и умение его обладателя работать с определенным программным или аппаратным обеспечением. Такая сертификация появилась в рамках программы Certified Novell Engineer (CME), запущенной в 1989 г. компанией Novell. Изначально она была нацелена на профессии, связанные с ИКТ, затем область ее действия переместилась на отдельные компании-производители и их отдельные продукты. С 1990 г. количество программ сертификации быстро увеличивалось, подобные программы появились и у других компаний-производителей программного и аппаратного обеспечения. К октябрю 2004 г. количество сертификаций превысило 2400, более чем от 70 компаний-производителей ИТ.

Однако сейчас используются также и второй вид сертификации, который направлен на освоение обучающимися некоторых принципов того или иного метода работы с ИКТ, например, объектно-ориентированного программирования или построения сетей. Примерами таких программ могут служить AMBI (основанная в 1960-х в Нидерландах) или современные A+ от CompTIA или IT Infrastructure Library (ITIL).

Но в целом, одной сертификации не достаточно для удовлетворения всех нужд. Рост числа программ сертификации при отсутствии единого центра развития, увеличивает запутанность самой сертификации. К тому же базовое образование также должно быть ориентировано на потребности бизнеса.

В целом состояние работ в данном направлении, несмотря на длительные период их проведения, можно охарактеризовать как находящееся в стадии разработки. Не существует более или менее универсального детализированного списка компетенций с их описаниями. Есть частные решения для отдельных компаний (например, «Аэробус»), отдельных стран (SFIA - Skills Framework for the Information Age, TAFE и др.), однако не существует единого стандарта разбиения области ИКТ на составляющие, не существует и алгоритмов перехода между уже существующими описаниями. Такое отсутствие единогласия, стандарта заметно замедляет развитие данного направления.

По вопросам управления знаний проводятся различные конференции, в том числе «European e-skills Conference», результатом которых

являются рекомендации, на основе которых можно выделить некоторые принципы построения системы компетенций.

Принципы построения системы компетенций

1. Для компетенций важно то, что реально приводит в конкретных условиях к улучшениям, а не то, что теоретически должно приводить к улучшениям в общем случае.

2. Компетенции делятся на уровни. Для каждого уровня владения компетенцией — свое описание, свой набор составляющих.

3. Согласно, Спенсерам, существуют компетенции пороговые (обязательные), без которых специалист не может считаться компетентным, а также дифференцирующие (дополнительные), определяющие качество работы.

4. Компетенции должны включать в себя не только профессиональные навыки. В состав компетенций входят:

- Кругозор (общие знания, абстрактные знания);
- Знания (конкретные профессиональные знания);
- Умения (практические знания);
- Навыки (уровень владения практическими знаниями);
- Личные качества.

5. Выделение компетенций и их составляющих выполняется эмпирически, на основании существующих профессий, требований и т. д.

6. Важно выделить целевую аудиторию, для которой используется система компетенций, и разделить выделить в ней группы, которые могут использовать для разных задач.

Результатом «European e-skills Conference» (сентябрь 2004 года) явилось разделение всех групп пользователей ИКТ на 3 группы:

- Профессионалы ИКТ (ICT practitioners), занимающиеся непосредственно ИКТ, т. е. те, для кого область ИКТ является и средством и целью работы. Это проектировщики программного и аппаратного обеспечения, программисты, персонал, устанавливающий и поддерживающий работоспособность программного и аппаратного обеспечения, а также занимающиеся управлением ИТ.
- Пользователи ИКТ (ICT users), для которых ИКТ — не конечная цель, а лишь средство достижения целей, не связанных с ИКТ — различные служащие; это самая большая группа.
- Руководители (e-Business), задача которых — инновация ИКТ, применение новых технологий для увеличения эффективности

и конкурентоспособности ведения бизнеса, в том числе — подбор соответствующих специалистов.

Однако данное деление представляется неполным. Предлагается другое деление, в котором дополнительно выделена группа людей, проводящих обучение,— инструкторов. Возможно также разделение группы профессионалов на неэксплуатационный (специалисты, занимающиеся разработкой) и эксплуатационный (специалисты, занимающиеся поддержкой) составы. Можно также выделить группу граждан (E-citizens). Это люди, которые практически не нуждаются в ИКТ в своей работе, однако используют их в повседневной жизни, в частности для общения.

Итого можно выделить следующие группы пользователей ИКТ:

- эксплуатационный состав профессионалов ИКТ;
- неэксплуатационный состав профессионалов ИКТ;
- пользователи ИКТ;
- руководители;
- инструкторы;
- граждане.

Примеры систем управления компетенциями

SFIA 3.0. Великобританская SFIA рассматривает общее деление на области знаний и умений (fields), необходимые специалистам в целом. Каждая из областей делится на подобласти — от 7 до 17, которые для SFIA являются компетенциями. Все области делятся на 7 уровней. Для каждого уровня каждой компетенции создано описание того, что должен делать специалист, обладающий заданной компетенцией. Основными областями в системе SFIA являются:

- стратегия и планирование (strategy & planning);
- развитие (development);
- изменение бизнеса (business change);
- предоставление сервиса (service provision);
- снабжение и поддержка управления (procurement & managment support);
- вспомогательные умения (ancillary skills).

В терминах SFIA профессия — это набор подобластей на определенных уровнях. Данная система не предусматривает создания «ролей» — соответствия набора компетенций профессиям, оставляя определение набора компетенций для описания профессии специалиста HR-менеджерам.

Как достоинство системы стоит отметить хороший подбор уровней:

1) следует (follow);
2) ассистирует (assist);
3) применяет (apply);
4) делает возможным (enable);
5) заверяет, советует (ensure, advise);
6) инициирует, влияет (initiate, influence);
7) определяет стратегию, воодушевляет, мобилизует (set strategy, inspire, mobilize).

Уровни полностью покрывают все ступени владения компетенциями. Названия легко понимаются и хорошо отражают суть уровня.

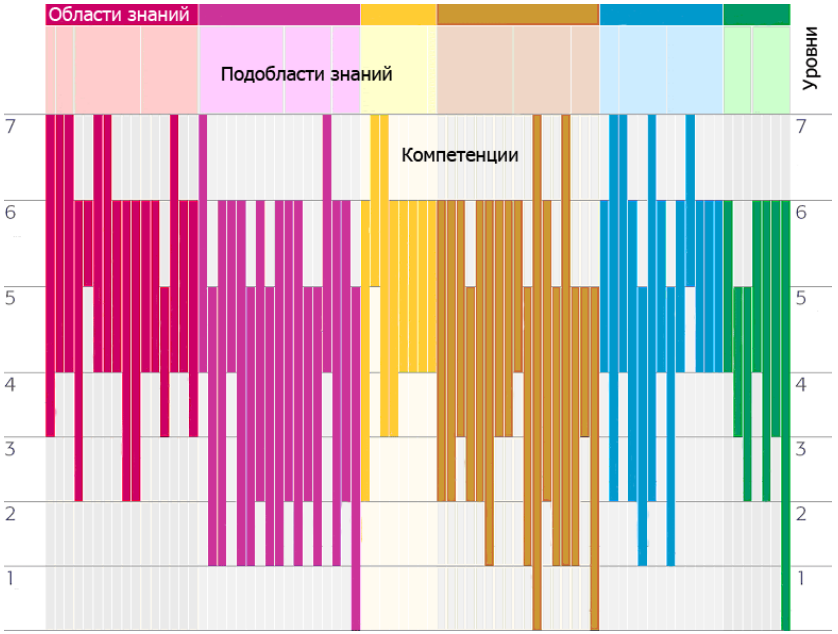


Рис. 1. Сводная таблица компетенций системы SFIA.

Все компетенции в рассматриваемой системе сформированы эмпирически, были уточнены в нескольких итерациях (версия системы — 3.0) и используются сейчас в Великобритании, поэтому можно использовать SFIA как источник, на основе которого выделять компетенции для своей системы.

Однако в SFIA можно выявить ряд недостатков, некоторые из которых представляются существенными при построении интегрированной системы управления человеческими ресурсами.

Во-первых, в ней не предусмотрены компетенции для знаний конкретного ПО, определенных принципов и технологий. В связи с этим систему нельзя назвать универсальной, т.к. она не позволяет ввести в систему сертификацию вендоров. Более того, она не имеет возможности описывать в виде набора компетенций знания узких специалистов, в частности, программистов.

Во-вторых, с точки зрения возможности привязки образовательных программ система также не идеальна, т.к. из-за некоторой расплывчатости формулировок знаний и умений разных уровней компетенций не все курсы можно четко к ним привязать.

В-третьих, SFIA не содержит ролей, а предлагает только набор компетенций, которые каждый HR-менеджер должен сам отбирать для своих вакансий.

TAFE (Австралия). TAFE выходит за пределы области ИКТ - в ее рамках составлена система компетенций также и по другим областям знаний: инженерии, финансам, спорту, благотворительности и т. д. TAFE ориентирована в первую очередь на поиск курсов по конкретным специальностям, поэтому вся структура компетенций направлена на реализацию этой практической задачи. В TAFE предусмотрены несколько градаций по уровням: формальное образование разбито на школьное (school sector) и университетское (higher education sector), а неформальное — на профессиональное обучение и курсы (тренинги) (vocational education and training sector). Три градации сопоставлены, и между их уровнями найдены соответствия (см. рис. 2).

Области знаний разбиты на отрасли, а они в свою очередь — на курсы по уровням и специальностям (двумерный массив). Фактически, компетенции выделяются внутри курсов как набор знаний, который эти курсы дают. Все курсы — производителенейтральные, т.е. дают знания в определенных областях, а не владение конкретным продуктом. Например, курс «Администрирование баз данных» (см. рис. 3) — Certificate IV in Information Technology (Database Administration).

Для курсов указаны входные знания (entry requirements) в виде общего описания, однако нет списков «входных» компетенций или «входных» составляющих компетенций. «Выходные» знания (required units) для всех компетенций строго разбиты на отдельные составляющие (unit), каждая из которых имеет свой код (code). Составляющие компе-

тенций уровней не имеют. Для компетенции указано также возможное название профессии — карьерные возможности (career possibilities).

School sector	Vocational Education and Training sector	Higher Education sector
		Doctoral degree
		Masters degree
	Vocational Graduate Diploma	Graduate Diploma
	Vocational Graduate Certificate	Graduate Certificate
		Bachelor Degree
	Advanced Diploma	Advanced Diploma
	Diploma	Diploma
Senior Secondary Certificate of Education	Certificate IV	
	Certificate III	
	Certificate II Certificate I	

Рис. 2. Таблица соответствия уровней градаций системы TAFE.

Course summary

Course detail	Description
Course summary	This qualification provides students with skills and knowledge in the administration of commercial database systems. Content areas covered include: OH&S, computer hardware/software and related documentation, determine and action network problems, provide one to one instruction, install software to network computers, complete database backup and records.
Study area	Information Technology
Entry requirements	This qualification may also be achieved through a traineeship. Completion of Year 10 or equivalent is recommended.
Career possibilities	Computing Support Technician

Required units

Unit	Code
Identify physical database requirements This unit details the competency required to create the physical database from the data dictionary and design specifications.	ICAITB060B
Monitor physical database implementation This unit deals the competency required to model and monitor database performance.	ICAITB061B
Create code for applications This unit describes the competency required to produce commercial grade program code and capture and handle errors which occur as part of the program operation.	ICAITB070B
Create user and technical documentation Define and document reference material to use, support and maintain system.	ICAITD128A
Apply skills in project integration Integration is the managment of overall project scope in the context of schedules, budgets, risk and contracts towards establishing agreed baselines for supplier/client requirements. Intergration involves the magement of the other eight functions of project management, making trade-off among competing objectives and alternatives in order to meet or exceed project objectives throught the project life cycle.	ICAITPM129A
Install and optimise system software This unit defines the competency required to apply aspects of systems optimisation.	ICAITS020C
Provide basic system administration This unit expresses the competency required to implement components of systems back-up, restore, security and licensing in a stand alone or client server environment.	ICAITS024C

Рис. 3. Пример описания курса в системе TAFE.

Система в целом показывает свою работоспособность, ее реализацию можно найти в Интернете по адресу http://www.tafe.qld.gov.au/. К плюсам TAFE можно отнести проработанность сопоставления уров-

ней компетенций и четкое выделение «выходных» составляющих компетенций для курсов.

Главным недостатком TAFE можно считать узкую направленность только на подбор курсов, следствием чего стала неразрывной связь курса с набором компетенций, которые он дает. Наличие нескольких градаций в системе в перспективе излишне, хотя на данный момент это строго отвечает системе образования. Кроме того, в системе отсутствует четкое деление «входных» компетенций.

GAHFA (Евросоюз). Cedefop — Европейский центр развития профессионального образования — с 1975 года занимается продвижением концепции обучения в течение всей жизни. Для возможности построения непрерывного образования, была построена система компетенций — GAHFA (European ICT skills framework). В ее рамках все знания в области ИКТ разделены на 4 области (Work Area), в двух из которых выделяются подобласти:

- коммерция и бизнес (commerce and business);
    - макретинг, консультирование и продажи (marketing, consulting and sales);
    - бизнес и управление проектами (business and project managment);
- приложения и администрирование (application and administration);
    - системы и разработка приложений (systems and application development);
    - бизнес и управление проектами (integration and administration);
- инфраструктура и установка (infrastructure and installation);
- поддержка и системный сервис (support and system service).

Области знаний делятся на поля деятельности (fields of activity), которые в данной системе можно рассматривать как компетенции. Каждая компетенция может иметь уровень от 2 до 6. Для полей деятельности существует набор заданий и умений (tasks and skills), один для всех уровней, но отличающихся для разных уровней качественно.

Не всегда возможно определить требования к профессии строго в пределах выбранных в данной системе компетенций, например, потому, что в небольших компаниях внутри одной профессии (вакансии) специалист должен выполнять больше работ в различных полях деятельности, а в крупных компаниях специализация более выражена.

В связи с этим в GAHFA было принято решение создать профили умений (skills profile) — один или несколько профилей в рамках области знаний, объединяющих все компетенции в данной области знаний.

Рис. 4. Структура системы компетенций GAHFA.

Именно для профилей умений (для каждого уровня — отдельное) составлено подробное описание, которое включает в себя:

- название и уровень профиля умений;
- пример названий профессий и существующих квалификаций в Европе, связанных с этим профилем;

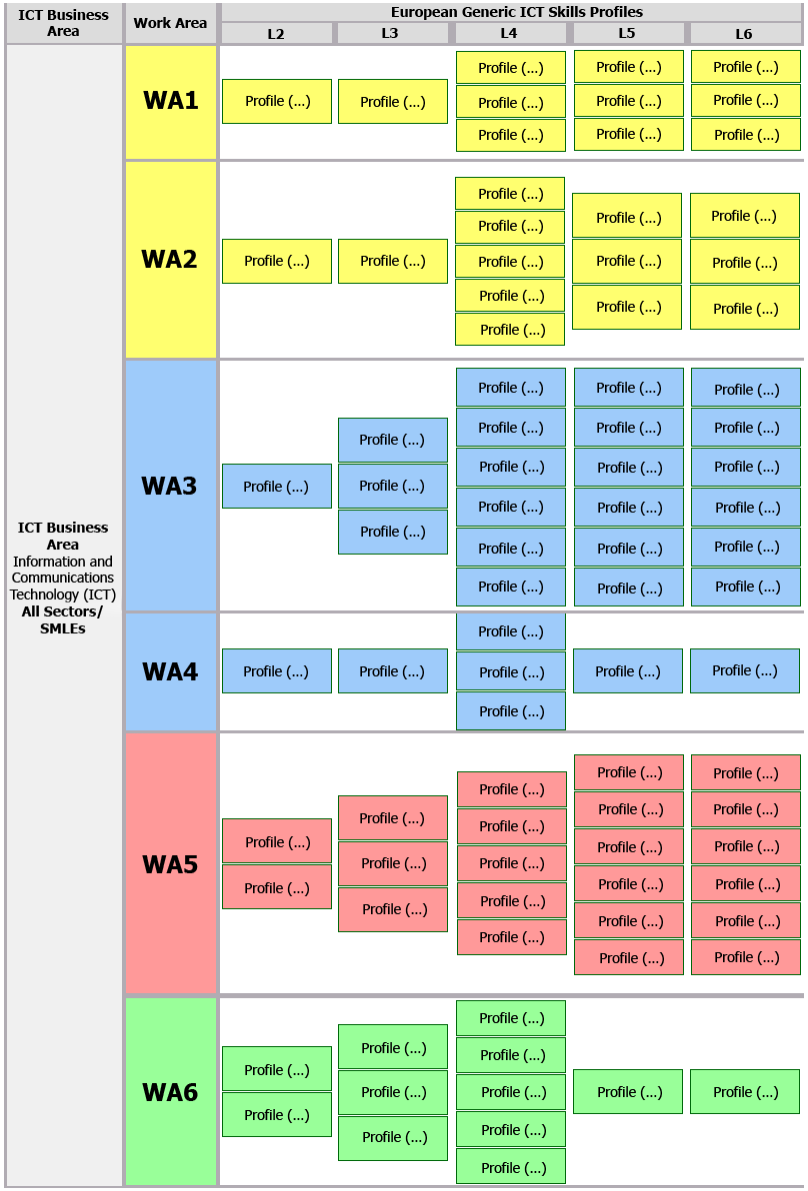


Рис. 5. Профили умений в системе GAHFA.

- род занятий и описание профиля;
- списки комплементарных знаний и умений в трех других областях знаний;
- карту карьеры и будущие возможности.

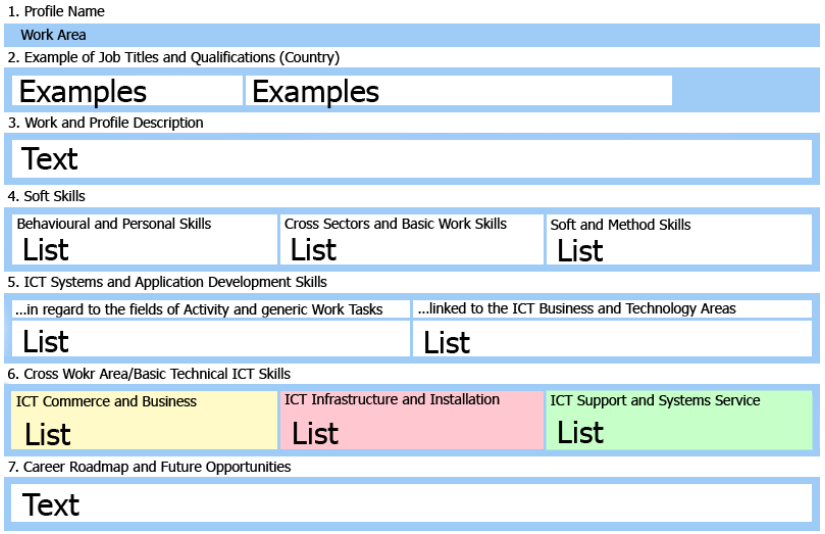


Рис. 6. Пример описания профиля умений в системе GAHFA.

К преимуществам системы можно отнести четкое описание профилей умений. Однако недостатком можно считать то, что деление здесь не древовидное, а сетевое: описание сделано не для отдельных компетенций, а для профилей умений. Профессии (вакансии) представляют собой не набор компетенций, а привязываются к профилю умений, причем четко не предписывается обладание конкретными компетенциями на заданной должности.

Разработанная система компетенций

За основу созданной нами системы была взята американская модель компетенций, использующая разбиение области знаний в ИКТ на 10 областей-ролей: Интернет и Интранет-технологии, информационная безопасность, операционные системы, поддержка пользователей, политика развития и планирование, программные приложения, сети и телекоммуникации, системное администрирование, системный анализ, управление данными.

Каждая роль имеет 4 уровня владения: начальный, средний, специалист, старший специалист.

Каждому уровню роли соответствует набор компетенций (технических или общих), каждая из которых — определенного уровня. Часть из компетенций является обязательными (ядровыми) для данной роли на данном уровне, а часть — рекомендуемыми. Каждая компетенция может быть одного из 5 уровней: базовый, фундаментальный, средний, опытный, эксперт.

Каждая компетенция на каждом уровне характеризуется набором поведенческих индикаторов. На рис. 7 представлена структура базы данных, реализующей модель. На ней:

1. Таблица Role — роль:
- ID — идентификатор;
- Name — название;
- Abbreviation — сокращенное название;
- Description — описание.

2. Таблица Atom — атомарная компетенция:
- ID — идентификатор;
- Name — название;
- Description — описание;
- Type — тип (общая, техническая).

3. Таблица Behavior — поведенческий индикатор:
- ID — идентификатор;
- AtomID — идентификатор компетенции, к которой относится индикатор;
- AtomLevel — уровень компетенции, к которой относится индикатор;
- Description — описание;
- Type — тип (кругозор, знание, умение, навык, личные качества, процессные компетенции).

4. Таблица Course — курсы;
- ID — идентификатор;
- Name — название по-русски;
- Number — код курса;
- Provider — провайдер курса;
- Description — описание;
- NameOriginal —название на оригинальном языке;

- ProductIT — продукт;
- Developer — разработчик курса;
- Price — стоимость;
- StudyForm — форма обучения;
- Language — язык, на котором преподается курс;
- GoalsObjective — цели и задачи курса;
- MainSubjects — главные темы курса;
- Program — содержание (программа) курса;
- Duration — длительность;
- TargetAudience — целевая аудитория
- RequiredCompetence — необходимые требования к кандидатам;
- OptionalCompetence — желательные требования к кандидатам;
- OutputCompetence — «выходные» компетенции;
- Link — ссылка на информацию о курсе в Интернете;
- Tests — связанные тесты;
- Sertificates — связанные сертификации;
- CoursesBefore — возможные предыдущие курсы;
- CoursesAfter — возможные последовательные курсы;
- CoursesAnalogs — аналогичные курсы;
- AdditionalData — дополнительная информация.

5. Таблица Role-Atom — связь ролей с атомарными компетенциями:

- RoleID — идентификатор роли;
- RoleLevel — уровень роли;
- AtomID — идентификатор атомарной компетенции;
- AtomLevel — уровень атомарной компетенции;
- Importance — важность (ядровая, рекомендуемая).

6. Таблица Course-Atom — связь курсов с атомарными компетенциями:

- CourseID — идентификатор курса;
- AtomID — идентификатор атомарной компетенции;
- CompLevel — уровень компетенции.

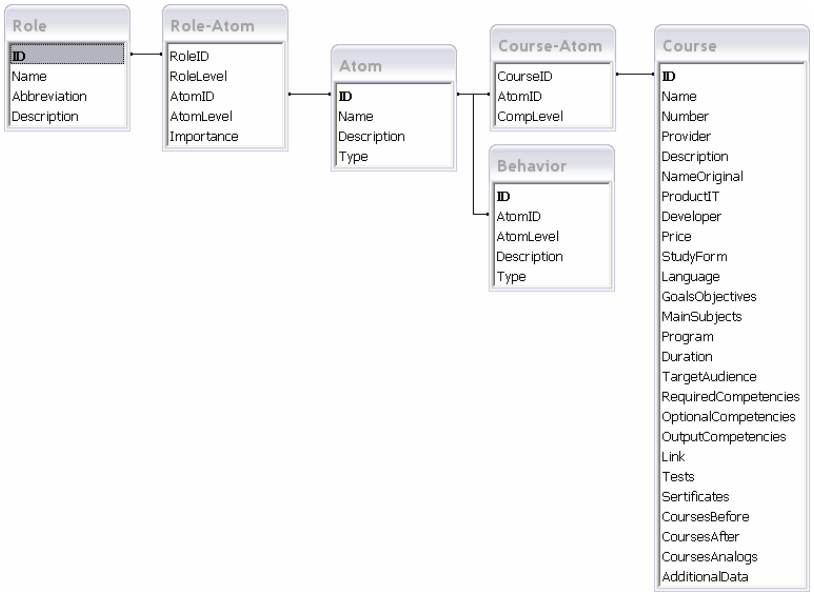


Рис. 7. Схема БД, реализующей созданную модель компетенций.

Анализируя созданную систему можно отметить следующие достоинства:

1. Легкая формализация структуры системы и представления ее для АСОИ. Частично это продемонстрировано в созданной БД, в которой не реализованы другие потенциальные возможности формализации — создание различных представлений для разных групп пользователей, использование разных видов поиска, создание экспертных систем для поиска.

2. Основными составными частями системы являются атомарные компетенции, причем ничто не ограничивает использование как производителезависимых, так и производителенейтральных компетенций.

3. В системе учтены различия между пороговыми (обязательными) и дифференцирующими (дополнительными) компетенциями для конкретной роли.

4. Описания компетенций структурированы — представляют собой набор поведенческих индикаторов.

5. Рассмотрены не только профессиональные знания, но и общие знания, и личные качества (в рамках компетенций и их компонентов).

6. Система легко расширяется: можно добавить роли, компетенции и поведенческие индикаторы, не нарушая существующей системы.

7. Система обладает гибкостью — существующие роли можно делать «базовыми», на основе которых строить роли (вакансии, профессии) для конкретных компаний. Базовые роли можно изменять в соответствии с новыми данными; устаревшие — убирать.

8. Достаточно проста реализация привязки существующих курсов и образовательных программ (в том числе сертифицирующих). Формализация позволяет выстраивать последовательности курсов и осуществлять поиск и подбор последовательности обучения на разных курсах с использованием ЭС. Таким образом, реализуется концепция обучения в течение жизни (lifetime education).

9. Данная система компетенций позволяет оценить перспективы карьерного развития.

Вместе с тем можно выделить и некоторые недостатки:

10. Рассмотрены компетенции только для тех, кто занимается ИКТ профессионально (эксплуатационный и неэксплуатационный составы), т. е. не учтены интересы следующих групп: пользователи ИКТ, руководители, инструкторы, граждане. Для учета всех групп пользователей необходимо выделить дополнительные роли.

11. Роль фактически является областью знаний, а это весьма неудобно применять на практике: название конкретной вакансии никогда не звучит как называние области знаний.

12. Уровни есть у ролей и компетенций, но названия у них разные, что потенциально может приводить к путанице. Лучшая градация уровней — в SFIA, хотя она, возможно, и требует уточнений — уровни названы по тому, что человек делает: следует, ассистирует, планирует и т. п. Там уровни полностью покрывают всю «вертикаль» знаний — от простого следования, повторения действий до стратегического планирования. В целом, система уровней требует доработки.

13. Не проработана система курсов: курс привязан только к компетенции безотносительно уровня компетенции. Однако плюс — курс привязывается к нескольким компетенциям (или несколько компетенций к одному курсу). Направление доработки — привязать к курсу компетенции на конкретных уровнях. Как следствие — курсы должны тренировать компетенцию на определенном уровне, а не отдельные поведенческие индикаторы (знания и умения).

14. Четко не описаны входные и выходные знания курсов в терминах модели компетенций. Это мешает, в частности, формализации и построению цепочек курсов и реализации других возможностей.

Предложения по улучшению разработанной системы компетенций

1. Глубину сети «роль»—«компетенция»—«поведенческий индикатор» необходимо увеличить до «область знаний»—«роль (вакансия, профессия)»—«компетенция»—« поведенческий индикатор». При этом значение «роли» отделяется от «области знаний», которая для роли служит лишь описательным атрибутом (между ролями и областями знаний связь «много—много»).

2. Необходимо осуществлять привязку курсов не просто к компетенции, а к компетенции на определенном уровне.

3. Для курсов необходимо указывать наборы входных и выходных компетенций. На начальном этапе это делать по возможности (т. к. не все современные курсы это позволяют сделать), а в дальнейшем — это требование переводить в разряд обязательных.

4. Необходимо расширить набор компетенций, введя дополнительные компетенции (и роли) для неучтенных групп пользователей — пользователей ИКТ, руководителей, инструкторов и граждан.

5. Предлагается ввести для компетенций систему уровней, такую же, как в SFIA. При этом для заданной компетенции не обязательно должны присутствовать все 7 уровней. А при отделении ролей от областей знаний отпадает необходимость градации по уровням областей знаний.

Литература

1. Лисицына Л. С. Теория и практика компетентностного обучения и аттестаций на основе сетевых информационных систем. — СПб: СПбГУ ИТМО, 2006. — 147 с.

2. Колесов В. П. О классификации компетенций // Высшее образование сегодня. 2006. № 2. С. 20—22.

3. Хуторской А. В. Ключевые компетенции и образовательные стандарты: Доклад на отделении философии образования и теории педагогики РАО, 2002. Центр «Эйдос». <www.eidos.ru/news/compet.htm>.

4. Байденко В. И., Оскарссон Б. Базовые навыки (ключевые компетенции) как интегрирующий фактор образовательного процесса // Профессиональное образование и формирование личности специалиста. Науч.-метод. сб. М., 2002.

5. Колер Ю. Обеспечение качества, аккредитация и признание квалификаций как контрольные механизмы Европейского пространства высшего образования // Высшее образование в Европе. 2003. № 3.

6. Интернет-источник: http://www.eucip.com.

7. Интернет-источник: http://www.sfia.org.uk.

К.Е.Иовков

ИСПОЛЬЗОВАНИЕ СРЕДСТВ СУБД ПРИ ПРОЕКТИРОВАНИИ И РЕАЛИЗАЦИИ КОМПОНЕНТ ЭКСПЕРТНОЙ СИСТЕМЫ*

Актуальность систем, основанных на знаниях, как одного из классов более общего понятия - интеллектуальных систем – со временем только набирает обороты. Так, Эдвард Фейгенбаум, основатель экспертных систем (а также разработчик ЭС DENDRAL - ЭС идентификации органических соединений с помощью анализа масс-спектрограмм, а затем и широко известной ЭС MYCIN, предметной областью которой является медицина), полагает, что, несмотря на свою высокую стоимость, разработка такого класса систем начинает со временем окупаться. Свидетельством этому факту является всё большая заинтересованность бизнеса в обобщении данных, получения статистических и концептуальных сведений о предметных областях, в которых осуществляется их функционирование.

Хранение и манипулирование данными, безусловно, являются ключевыми задачами большинства автоматизированных систем; при этом способы хранения данных (например, с использованием объектных, реляционных или файловых моделей хранилищ данных) так же разнообразны, как и принципы управления данными. На сегодняшний момент известно множество СУБД, проверенных временем и оправдавших себя в качестве надёжных и функциональных систем, имеющих известные преимущества и, как и всё существующее, компромиссно обладающие своими недостатками. Это позволяет достаточно легко сделать выбор и разработать эффективные структуры данных и снабдить их необходимой логикой. И если на этапе становления таких классов систем стояла задача создания адекватной "линейной" обработки данных и способа их хранения, то с ростом объемов информации наиболее обещающей с точки зрения значимости результатов стала задача анализа уже накопленной информации (так, по данным компании IBM, всего за последние 5 лет объем такой информации в базах данных крупных компаний вырос примерно в сто раз [7]). С развитием необходимых средств

---

* Научный руководитель к.т.н., доц. А.Ю.Филиппович. Работа выполнена в 2006–07 уч.г.

появилась возможность использовать накопленные знания[6] для их дельнейшего использования в качестве некоторого необходимого ресурса для формирования выводов. Использование результатов такого анализа позволяет осуществлять поддержку принятия решений, в значительной степени автоматизировать производство или осуществлять необходимый мониторинг. Все эти задачи решаются с помощью систем моделирования: от имитационного – на нижнем уровне, до ситуационного – на верхнем [3].

Основополагающей системой с точки зрения анализа данных является экспертная система, или система экспертного моделирования. Однако, как уже было отмечено ранее, дороговизна таких систем, трудоёмкость и некоторая степень недоверия (пусть и очень малая в случае высококлассных ЭС) к результатам их работы, может свести на "нет" все положительные аспекты их использования: нет смысла вкладывать значительные ресурсы в разработку сложной системы, когда есть большая вероятность того, что затраты "покроют" все выгоды. Таким образом, если есть некоторая необходимость разработать среду для анализа показателей каких-либо датчиков, то и затраты на реализацию такой системы должны быть минимальными. Одним из способов достижения этой цели и является разработка всех компонентов ЭС как отношений в единой схеме базы данных и программирование логики вывода с использованием доступных средств СУБД.

Реляционные СУБД и их расширяемая функциональность.

Одним из основных компонентов, необходимых для функционирования ЭС, является база знаний [4], содержащая факты предметной области (декларативные знания) и правила вывода (процедурные знания). В общем случае база знаний предназначена для хранения долгосрочных данных, в т.ч. и правил, а база данных – исходных и промежуточных данных, используемых для осуществления вывода (последовательного получения новых данных, на основе которых на некотором шаге может быть принято окончательное решение, являющееся целью функционирования экспертной системы). Однако, вводя разграничение функциональностей базы данных и базы знаний, приходится констатировать возникновение проблемы интеграции базы данных и базы знаний. Используя различные источники данных и процедурные знания, необходимо поддерживать совместимость и доступность каждого из источни-

[6] В данном случае имеется в виду структурированные данные, из которых затем могут быть получены знания в контексте определений интеллектуальных систем

ков, а также обеспечивать непрерывность канала передачи данных. Только в случае использования единой среды функционирования удаётся минимизировать данную проблему (однако использование единой СУБД предпочтительней, чем использование файловой системы, когда, в последнем случае, возникает задача реализации чёткого управления структурой файла – например, может возникнуть дублирование данных из-за децентрализованной работы с данными, зависимость от структуры данных и типа используемых файлов и т.п. [1]). Более того, использование разнообразных источников приводит к увеличению стоимости разработки и поддержки системы, что приемлемо только в случае, если разрабатываемая функциональность должна обладать высочайшими характеристиками (например, ЭС G2 или MYCIN), а это не является задачей первичной важности в ситуации уменьшения затрат на разработку системы.

Далее, выделив процедурные знания и определив форму их представления, появляется несколько альтернатив реализации машины вывода. Существует несколько моделей представления знаний (логические, продукционные, сетевые, фреймовые и т.д.), в соответствии с которыми необходимо реализовать определённую логику интерпретатора. Для этого можно использовать определённый язык моделирования систем [3]; однако нет реальной необходимости использовать нечёткий вывод или семантические сети для решения задачи реализации, например, интеллектуального помощника системным администраторам или системы мониторинга показателей квазидинамических или статических систем – достаточно организовать процедурные данные в простой форме правил "если-то" и т.п. Поэтому использование сложных языков (таких как FRL (Frame Representation Language, фрейм-ориентированный язык), OPS (продукционный язык, использующийся в системе R1), KRL (Knowledge Represantation Language), ARTS и т.п. [6]) или, более того, разработка нового, в ряде случаев не оправдано. Кроме того, написание интерпретатора языка или заимствование его у сторонних разработчиков порождает ряд трудностей, связанных с увеличением трудоёмкости или стоимости соответственно. Совершенно иначе дело обстоит со стандартными языками, использующимися повсеместно (например, C++, Java или более специализированными языками Lisp, Prolog). Более того, если данный язык является частью одной системы, содержащей все знания о предметной области (ПО), то уменьшается сложность реализации интерпретатора как распознавателя языка.

Пусть теперь необходимо разработать адекватную и наиболее общую модель, позволяющую реализовывать как автономные, так и гиб-

ридные экспертные системы для различных предметных областей, исходя из следующих ограничений:

1) Необходимо реализовать либо статическую, либо квазидинамическую (т.е. не динамическую) систему экспертного моделирования;
2) Предметная область (или области, для к. ведётся разработка) обладает свойством структурной определённости представленных в ней объектов, т.е. можно установить достаточно чёткие правила вывода (эта позиция объясняется далее при рассмотрении структуры объектов ПО);
3) Приняты во внимание все вышеперечисленные утверждения о наиболее подходящей структуре и реализации БЗ, БД, языка и машины вывода как следствие минимизации трудоёмкости и затрат на разработку.

Анализ данных ограничений позволяет сделать заключение о том, что в таком случае наиболее подходящей системой, обладающей собственной функциональностью, языком и формой хранения данных, является СУБД. Системы такого класса способны эффективно манипулировать большими объёмами данных и позволяют извлекать и обрабатывать информацию с использованием достаточно развитых методов структурированного языка запросов.

Формализованная модель систем экспертного моделирования

Для осуществления основных функций системы экспертного моделирования введём общее понятие системы как совокупности объектов и принятой модели вывода:

$$\Omega = \langle O, M \rangle, \quad (1)$$

где: $O = \{o_i\}$ - множество однотипных объектов; $o_i = \{[at]_j\}$ - отдельный объект, представляющий собой множество атрибутов, где $[at]_j$ - отдельный атрибут. В данной системе выбор объекта является целью экспертного моделирования.

Будем полагать, что объект o_i представляет собой объединение

$$o_i = A \cup P, \quad (2)$$

где: $A = \{a_i\} \subset \{[at]_j\}$ - селективный объект (элементы этого объекта a_i будем называть селективными атрибутами);

$P = \{p_i\} \subset \{[at]_j\}$ - параметрический объект (элементы этого объекта будем называть параметрами), причём

$P \bigcap A = \varnothing$

При этом отличие параметрического объекта от селективного состоит в том, что элементы второго (селективные атрибуты) не меняются с течением времени моделирования. Поэтому состояние [sn] системы Ω в конкретный момент времени τ есть множество комбинаций параметров и их значений

$$[sn](\Omega) = \left\langle \left\{ p_i, [const]_j \right\}, \tau \right\rangle, \quad (3)$$

где: $p_i \in P$ - параметр, а $[const]_i \in [CONST]$ - значение данного параметра из множества констант, не зависит от селективного объекта. Тем ни менее, селективные атрибуты являются важной составляющей объекта: значения селективных атрибутов (которые также являются элементами множества констант [CONST]) представляют собой некоторые критерии, характеризующие сам объект: они позволяют выделить этот объект из множества других.

Далее для формулы (1):

M - общее понятие модели вывода;

$$M = \left\langle R, S^M \right\rangle, \quad (4)$$

где: $R = \{r_j\}$ - множество правил (rules), использующихся для вывода; число j называется номером правила r_j, $j \in N^P$. N^P – конечное множество допустимых номеров правил;

S^M – синтаксические правила языка, позволяющие строить из правил R логически верные конструкции.

Общий вид правила r_j для любой модели будет определяться как

$$r_j = \left\langle L, f, S, C, H \right\rangle, \quad (5)$$

где: L (logic) – условие, составленное по синтаксическим правилам S. Истинность определяется функцией истинности $f(L_i) = \{true, false\}$, где $L_i \in L$;

$f \subset F$ - подмножество множества операторов и функций F, например

F = {+,-,*, /, |, &, ^, <=, >=, =, !=, ...}

S (syntax) – синтаксис, позволяющий составлять условные выражения из элементов f и L;

C (conclusion) – заключение, вытекающее из истинности условия (посылки);

H (characteristic) – множество индивидуальных свойств правила.

Учитывая, что ранее было наложено ограничение, связанное с тем, что процедурные знания представлены в форме if-правил, то можно ввести понятие множества команд I (instruction), позволяющие осуществлять изменение состояния системы и представляющее собой множество упорядоченных пар:

$$I = \left\{ p, \hat{V} : p' =' \hat{V} \right\}, \quad (6)$$

где: $p \in P$ - параметр;

$$\hat{V} = \left\langle v, f^V, S^V \right\rangle, \quad (7)$$

где: $v \subset V^P$ – элементы множества $V^P = (P \cup [CONST])$ - объединения параметрического объекта и множества значений;

$f^V \subset F$ - подмножество множества операторов и функций, куда не входят операторы сравнения, например

$f^V = \{+,-,*, \backslash, |, \&, ^\wedge\}$;

S^V – синтаксис, позволяющий строить из элементов V с использованием операторов f^V выражения;

$'='$ - оператор присваивания.

Таким образом, множество команд есть $I = \left\{ [ins]_j \right\}$, $j \in N^I$, где N^I – множество допустимых номеров команд. Отдельная команда будет иметь вид $<p_a> = <p_b|v_b> [<f_i^V> <p_c|v_c>]$. Примеры команд:

<p1 = 30 + p2>
<name = ‘john’>
<p3 = p4 + p5 * p6>

Правила и модели вывода

Определив основные понятия и исходя из классификации экспертных систем по взаимодействию с другими системами моделирования, можно выделить следующие модели вывода:

$M^A = \left\langle R^A, S^{MA} \right\rangle$ - модель вывода автономной ЭС

$M^H = \left\langle R^H, S^{MH} \right\rangle$ - модель вывода гибридной ЭС

Несмотря на вполне определённую структуру, формализованные модели могут содержать различные типы правил и синтаксиса. При этом можно выделить несколько типов правил, которые вместе составляют

некоторый базис и, в соответствии с определёнными ранее понятиями и структурами, должны быть реализованы в любом из вышеперечисленных типах систем:

α - правила - правила изменения состояния,

γ - правила - правила выбора селективного объекта.

Именно эти типы правил обязательны потому, что в данной модели системе Ω определены 2 типа объектов, обработка которых является ключевой задачей ЭС. Этот список может быть расширен в зависимости от того, какая стратегия вывода должна быть заложена в конкретной разрабатываемой модели.

Физическая реализация и связь с реляционными структурами данных

После составления формализованный модели можно определить, где и каким образом возможно реализовать рассмотренные выше структуры.

Объект. Селективный объект – это некоторый объект предметной области, содержащий атрибуты с определёнными значениями. Количество атрибутов, в общем случае, может меняться, но в реальных задачах выбора объекта чаще всего используется конечный набор атрибутов с постоянными значениями. Поэтому, жёстко зафиксировав структуру, или, говоря языком реляционной алгебры, определив схему отношения на множестве атрибутов [1,2], можно реализовать множество таких объектов в качестве экземпляра отношения, или таблицы, где объект представляет собой кортеж (строку) экземпляра отношения (таблицы).

Параметрический объект представляет собой объект, который, в общем случае, также имеет набор атрибутов и их значений, меняющихся во времени. Значения атрибутов такого объекта требуют инициализации для каждого нового процесса моделирования и изменяются правилами вывода в процессе моделирования; на каждом шаге вывода набор таких значений определяет состояние системы. Вводя ограничение на фиксированное количество атрибутов параметрического объекта и их структуру, аналогично можно определить и схему отношения. В таком случае, используя DML – синтаксис структурированного языка запросов, можно легко изменить состав и количество объектов.

Рассматривая класс автономных экспертных систем, необходимо также реализовать и структуры данных, позволяющие хранить вопросы по предметной области, а также ответы на такие вопросы. Реляционные СУБД позволяют достаточно легко построить схемы отношений "вопрос", "ответ", "подсказка" и т.п., включая схемы отношений, позво-

ляющие динамически задавать параметры интерфейса при осуществлении диалога с пользователем (самый простой вариант – это указать интерфейсу тип и количество вопросов), представляя информацию в естественной и наиболее удобной форме. Связав все схемы отношений по 3 нормальной форме и установив ограничения (constraints), получим неизбыточные и непротиворечивые данные, при этом минимизировав трудоёмкость данных операций.

Необходимо также учитывать, что реляционная алгебра и её реализации требуют уникальности имён атрибутов [1,2] для всех схем отношений.

Более сложным вариантом реализации ЭС будет наличие в структуре селективного объекта таких атрибутов, значения которых могут быть оценены только субъективно или с использованием статистики, а также другими методами анализа данных. Однако в таком случае с помощью процедурных знаний, представленных в виде простых if-правил, достаточно сложно адекватно оценить значения вышеуказанных атрибутов, так как в таком случае появится необходимость использования, например, статистических данных, или же реализации других форм вывода (например, с помощью нечётких множеств).

Правило. Как уже было описано выше, для всех систем должны быть определены 2 вида правил: α и γ. Если разрабатываемая ЭС принадлежит классу автономных систем, то необходимо реализовать функциональность, позволяющую выбирать подходящие вопросы из некоторого хранилища, поэтому модель вывода для автономной системы должна также включать и правила выбора следующего вопроса (β - правила), т.е. $R^A = \{\alpha\} \cup \{\beta\} \cup \{\gamma\}$.

Согласно предложенной модели, правила, как и объекты, обладают чёткой структурой. Следовательно, спроектировав адекватную схему отношения, для правил также можно использовать реляционную форму представления. При этом экземпляры схем таких отношений не будут связаны с другими экземплярами, но будут зависеть от их структуры (например, условие правила выбора селективного объекта зависит от того, какие названия имеют атрибуты данного объекта). Существует и другой вариант, когда атрибуты объекта именуются по определённой схеме (например, названию атрибута присваивается номер атрибута в схеме отношения (т.е. столбца таблицы)), но данный вариант более сложен с точки зрения разработки логики функционирования машины вывода.

Согласно общему определению правила (см. формулу (5), оно содержит некоторое условие и посылку). При этом очевидно, что в какой бы форме не хранились процедурные знания, необходимо использовать некоторый синтаксический анализатор и исполнитель команд, записанных в поле посылки правила. Таким образом, в общем случае необходимо реализовать 2 модуля: синтаксический модуль распознавания условия и посылки и исполнительный модуль. Но, учитывая ограничения, введенные в выше, желательно избежать такого разделения. Это возможно, как и в случае управления объектами при использовании структурированного языка запросов.

Синтаксис, набор функций и операторов. Как уже описывалось ранее, существует необходимость проверки условий правила вывода и выполнения посылки. Так, если в любом из правил поле условия будет записано в текстовой (!) форме в терминах предиката языка SQL (WHERE <условие>), а посылка – в форме <атрибут = значение> [,..n], то достаточно будет просто выполнить запрос DML

EXISTS (SELECT * [,..n] FROM <таблица объектов> WHERE <условие>)

для проверки условия посылки, и

UPDATE <таблица объектов>
SET <атрибут = значение> [,..n]
WHERE <условие>

для изменения значения атрибутов объекта. При этом операторы, которые можно использовать в предикате, позволяют проверять достаточно сложные условия, а посылка может включать в себя как установку параметров (параметром здесь и далее называется атрибут параметрического объекта) равными константам, так и использовать выражения с другими параметрами.

Негативным следствием такого подхода является появление вполне логичной структурной зависимости между правилами и объектами, что достаточно сильно усложняет разработку правил вывода и их внесение в базу данных. Однако минимизировать влияние такой зависимости можно разработав графический или иной интерфейс, позволяющий редактировать и добавлять правила и атрибуты одновременно, что является стандартной практикой при разработке любой ЭС (с другой стороны, возможность редактирования правил непосредственно в базе данных можно отнести и положительным моментам, так как это позволяет не разрабатывать низкоуровневый редактор правил).

Главным достоинством данного подхода является то, что он позволяет полностью исключить разработку или интеграцию модуля распознавателя и проверки синтаксиса, а также использовать доступные средства взаимодействия с СУБД как основу логики интерпретатора.

Пример физической реализации реляционной модели экспертных систем

Рассмотрим физическую структуру базы данных и вариант реализации машины вывода на примере системы Query Expert Designer (QED). Данная система обладает графическим интерфейсом и позволяет создавать автономные и гибридные квазидинамические экспертные системы. Гибридные ЭС представлены двумя моделями вывода (N – моделью (содержит α- и γ- правила) и F – моделью (содержит α-, γ- и специальные δ- правила)).

1. Реализация схемы отношения "селективный объект"

Согласно формализованной модели системы, представленная схема отношения Objects есть селективный объект (см. формулу (2)).

Поле	Тип	Значение
key_object	Int	ключ (ID) объекта
Active	Bit	активность
_rowcount	Int	число выборки объектов
<attributes>	любой допустимый тип TSQL	селективные атрибуты

Кроме фиксированных трёх полей, объект содержит множество атрибутов – столбцов со своими уникальными именами.

В дальнейшем, «объектом» будем называть определённую запись (строку) таблицы Objects, а «атрибутом объекта» - поле (пересечение строки и столбца) данной таблицы; имя атрибута определяется как название столбца.

2. Реализация схемы отношения "параметрический объект"

Поле	Тип	Значение
key_params	Int	ключ (ID) объекта
<attributes>	любой допустимый тип TSQL	параметрические атрибуты

В данной схеме отношения (Params) значение <attributes> также представляет несколько полей с фиксированными неповторяющимися названиями. На каждую сессию работы ЭС (т.е. для каждого её запуска)

в эту таблицу добавляется одна строка, ключ который хранится системой. Поэтому, изменяя параметр с определённым именем, система фактически изменяет поле данных в столбце с таким же именем; номер строки при этом определяется по её известному ключу.

3. Ддополнительные схемы отношений для автономных ЭС

В автономной модели используется режим советчика – последовательно задаётся ряд вопросов, однозначно связанных с ответами из таблицы ответов, и одной текстовой подсказкой. При этом могут использоваться ответы различных типов: «вариант ответа», «текст» и «диапазон вводимых значений».

1) Схема отношения "вопрос" (Questions):

Поле	Тип данных	Значение
key_question	Int	Ключ
Text	nvarchar(200)	текст вопроса
Class	Bit	тип вопроса
answer_type	char(1)	тип ответа
Active	Bit	свойство активности

Таблица вопросов содержит текст вопроса и тип ответов на данный вопрос (тип указывается одним символом – ‘R’ (варианты ответов), ‘T’ (текст, вводимый пользователем), ‘V’ (диапазон вводимых значений)), используемые интерфейсной частью программы. Каждый тип ответа предполагает особую структуру непосредственно данных из полей записей таблицы ответов. Свойство *активности* используется при выборе следующего вопроса из таблицы вопросов (может быть задан только вопрос с единичным значением этого поля); свойство *class* представляет тип ответа, и делит все вопросы на 2 класса: стартовые и промежуточные. Стартовым является вопрос, содержащий ‘0’ в поле ‘class’. Такие вопросы независимы, их можно задать в любой момент времени. Во время старта программы выбирается первый из попавшихся таких вопросов. Второй тип вопросов – промежуточный – выбирается уже на основе β - правил.

2) Схема отношения "ответ" (Answers):

Поле	Тип данных	Значение
key_hint	Int	Ключ
key_question	Int	внешний ключ вопроса
Text	nvarchar(50)	текст ответа
param_values	varchar(100)	командное выражение

Каждый ответ на вопрос представлен текстовым полем, т.е. текстом ответа; полем для связи с таблицей Questions (key_question, для установления соответствия конкретного ответа вопросу), тип связи – много к 1 соответственно, т.е. одному вопросу может соответствовать несколько ответов. Поле param_values содержит командное выражение, описанное в формуле (13), указывающее, какому параметру какое значение должно быть присвоено в случае ответа на данный вопрос.

Если ответ имеет тип ‘T’ (текст) (тип – это поле answer_type таблицы Questions - определяется по связи), то обязательно при этом наличие одного варианта ответа на данный вопрос, где значение поля “text” (т.е. непосредственно текста ответа) начинается с символа ‘*’:

*<текст ответа>.

Такой вариант ответа считается вариантом ответа по умолчанию и предлагается пользователю в случае, если введённый им текстовый ответ не совпадает ни с одним из возможных вариантов ответа.

Если ответ имеет тип ‘V’ (диапазон значений), то в поле “text” заносится строка вида:

“<значение1>, <значение2>”,

где: <значение> - конкретное число.

Если ответ имеет тип ‘R’ (вариант), то в поле “text” должен быть указан обычный текст варианта ответа: <текст ответа>.

4. Схема отношения "α - правило" (правило изменения состояния)

α - правила позволяют изменять значения определённого множества параметров в зависимости от истинности логического условия, также содержащегося в правиле, и их выполнение осуществляется с использованием стратегии разнообразия: при истинности такого условия атрибут активности правила сбрасывается, и выполняется посылка. Переменными посылки могут являться только параметры.

Выполнение посылки предполагает изменение значений ряда параметров, список имён которых указан в поле “param_values” (поэтому такие правила называются правилами изменения состояния системы). В случае обработки α - правил автономной модели также сбрасываются атрибуты активности вопросов из таблицы Questions, список идентификаторов (ключей) которых перечисляется через запятую в поле с именем “inactive_questions”.

1) Схема отношения "α - правило" ЭС гибридного класса (CondRules):

Поле	Тип данных	Значение

key_rule	int	идентификатор правила
Active	int	свойство активности
Condition	varchar(150)	условное выражение
param_values	varchar(100)	командное выражение

2) Схема отношения "α - правило" ЭС автономного класса (CondRules):

Поле	**Тип данных**	**Значение**
key_rule	int	идентификатор правила
Active	int	свойство активности
Condition	varchar(150)	условное выражение
param_values	varchar(100)	командное выражение
inactive_questions	nvarchar(70)	список исключаемых вопросов

5. Схема отношения "β - правило" (правило выбора команды)

Наличие β - правил характерно только для автономных ЭС. Такие правила содержат условное выражение, синтаксически аналогичное выражению α - правила, и поле “next_question_id”, содержащее идентификатор (ключ) вопроса из таблицы вопросов. При истинности условия атрибут активности правила также сбрасывается, и выполняется посылка – интерпретатор получает команду для вывода вопроса с соответствующим полю “next_question_id” идентификатором.

Использование β - правил также представляет реализацию стратегии разнообразия.

Схема отношения "β - правило" (JmpRules):

Поле	**Тип данных**	**Значение**
key_rule	Int	идентификатор правила
Active	Int	свойство активности
Condition	varchar(150)	условное выражение
next_question_id	Int	Ключ следующего вопроса

6. Схема отношения "γ - правило" (правило выбора объекта)

Данный тип правил имеется в каждой из 3-ёх моделей вывода и имеет полностью идентичную структуру. По сравнению с другими правилами, такая структура является наиболее сложной, что определяется необходимостью осуществления главной цели вывода – выбора объектов из соответствующей таблицы.

При истинности логического условия выполняется посылка (поле “param_values_if”). Посылка данного типа правил содержит некоторые критерии, на основании которых выбираются уже непосредственно объекты. Таким образом, посылка также представляет собой условное выражение, но в этом случае переменными такого выражения являются уже не параметры, а имена атрибутов объектов. Имена параметров в качестве переменных недопустимы, так как в данном поле составляется условие выбора селективного объекта из соответствующей таблицы, а не параметра из таблицы параметров. Однако существует возможность вставки значения одного из параметров в качестве константы, с которой сравнивается переменная-атрибут: в таком случае в поле посылки “param_values_if” символ ‘*’ заменяется значением параметра, имя которого указано в поле “param_name”. Такого вида посылка нужна для того, чтобы потом, объединив в одно логическое условие все посылки из “param_values_if”, для которых выполнилось условие "condition", выбрать подходящий объект (если необходимо составить условие на основе нескольких параметров, то нужно записать несколько правил с одинаковым полем "condition" и необходимыми параметрами, содержащимися в поле "param_name").

Второй атрибут – поле “basic” - представляет приоритет правила. Оно используется при выборе очередного правила данного типа машиной вывода: чем выше приоритет правила, тем позже оно будет обработано.

В соответствии с приоритетным механизмом выбора правил, стратегию такого вывода можно назвать стратегией предпочтения.

Схема отношения "γ - правило" (ChoiceRules)

Поле	**Тип данных**	**Значение**
key_choice_rule	Int	идентификатор правила
Basic	Bit	приоритет
Condition	varchar(200)	условное выражение
param_name	varchar(20)	имя параметра замены
param_values_if	varchar(100)	посылка

7. Схема отношения "δ- Правило" (правило завершения вывода)

δ - правила используются только в F-модели вывода гибридных ЭС. Данный тип правил представлен стратегией фокусирования – обрабатываются только те правила, которые были активированы (установлен атрибут активности) при совершении события – изменении определённого параметра или их комбинации; а также стратегией разнообразия.

Обработка правил предполагает проверку логического условия, синтаксис которого не отличается от синтаксиса условий α- и β- правил, и при его истинности процесс вывода останавливается.

Физически данный тип правил обрабатывается триггером, срабатывающим при модификации определённых параметров и осуществляющим механизм активации правил. Поле “param_name” правил содержит специфический номер (данный номер рассчитывается как десятичное представление двоичного числа, где в 1 устанавливается бит с номером, равным номеру столбца-параметра в таблице параметров Params, если правило должно быть выполнено при изменении данного параметра), позволяющий связать активацию данного правила с изменением параметра или их комбинации.

Также как и во всех предыдущих типах правила, при выполнении логического условия выполняется посылка – в данном случае она формальна, и выполнение посылки любого δ- правила сигнализирует о том, что вывод должен быть остановлен.

Схема отношения "δ - правило" (FinalRules)

Поле	**Тип данных**	**Значение**
Key_rule	Int	идентификатор правила
Active	Bit	свойство активности
Condition	varchar(100)	условное выражение
param_name	Int	номер активации

8. Логика вывода

Рассмотрим упрощённый вариант функционирования интерпретатора.

8.1 Алгоритм вывода для модели класса автономных ЭС

1) Получение стартового вопроса (т.е. вопроса с нулевым значением поля class). Если такого нет, или они все не активны, то вывод считается завершённым;

2) Получение ответов на вопрос, выбор ответа пользователем. Каждый ответ предполагает выполнение командного выражения (т.е. изменение одного или нескольких параметров, указанных в поле param_values) этого ответа. В результате этого изменяется состояние системы;

3) Любое изменение состояния системы может быть «цепным», т.е. при изменении одного параметра могут (и должны) измениться другие параметры. Поэтому после выполнения командного выражения

производится анализ таблицы CondRules (α - правила) и просмотр и выполнение активных α - правил. Если хотя бы одно α - правило было выполнено, то оно деактивируется (поле active сбрасывается в 0) и выполняется посылка (деактивируются вопросы и изменяются параметры). После анализа остальных α - правил, таблица будет просмотрена заново – такие циклические просмотры происходят до тех пор, пока не окажется, что условия всех активных правил не выполняются. Это значит, что состояние системы далее измениться уже не может, и необходимо выполнить следующий этап – поиск вопроса, выбор которого зависит от текущего состояния системы.

Физически для каждого выбранного таким образом α - правила создается специальный запрос вида

```
UPDATE Params SET <param_values>
WHERE <condition>
```

где

Params – имя таблицы параметров;

<param_values> - значение, которое стоит в поле param_names анализируемого правила;

<condition> - значение, которое стоит в поле condition анализируемого правила.

Данный запрос совмещает в себе сразу две функции – проверку условия и выполнение посылки.

4) Анализ β - правил из таблицы JmpRules. Таблица просматривается один раз, и при выполнении условия некоторого правила, его свойство активности сбрасывается, и поступает команда на выборку вопроса, ключ которого находится в поле next_question_id. Если не один промежуточный вопрос не был выбран, то выбирается первый активный стартовый вопрос. Только в случае, если ни один вопрос не выбран или в базе знаний не содержится активных правил, система переходит к выполнению γ - правила.

Для проверки логического условия, содержащегося в поле condition каждого активного β - правила, создаётся запрос на пустую выборку, и проверяется возможность его выполнения:

```
SELECT … FROM Params WHERE <condition>
```

5) Выбор объектов в зависимости от значений изменённых (в процессе получения ответов) параметров. На первом этапе все правила сортируются по убыванию приоритета. Затем, при истинности условия γ - правила, производится инкремент поля _rowcount (число выборов объ-

екта) для всех объектов из списка объектов (в начале моделирования такой список состоит из множества ключей всех объектов базы).

а) Если правило с текущим приоритетом одно, то создается запрос следующего вида на выбор объектов:

```
UPDATE dbo.Objects
SET _rowcount = _rowcount+1
WHERE active=1
AND <condition>
AND key_object IN (<key_list>)
```

где вместо <condition> будет подставлено поле param_values_if текущего правила, а вместо <key_list> - список объектов, которые были выбраны при выполнении предыдущего правила.

б) Если правил с текущим приоритетом несколько, то из условий выборки каждого из таких правил (т.е. из полей param_values_if) будет скомпоновано одно общее условие, являющееся их дизъюнкцией. Такое общее условие и будет подставлено в запрос:

```
UPDATE dbo.Objects
SET _rowcount = _rowcount+1
WHERE active=1
AND (<condition_1> OR <condition_2>
OR … OR <condition_n>)
AND key_object IN (<key_list>)
```

где

<condition_1> … <condition_n> - значения полей “param_values_if” каждого из правил, имеющих одинаковый приоритет, и, аналогично, вместо <key_list> - список объектов, которые были выбраны при выполнении предыдущего правила.

Как видно, каждая последующая выборка осуществляется с учётом результатов предыдущей, т.е. число объектов с каждым выбранным правилом уменьшается (или, точнее, не увеличивается) – каждое выполненное правило способно выбирать только из тех объектов, которые были выбраны предыдущим γ - правилом (или, если это правило первое, то из всех доступных объектов).

6) Процесс вывода продолжается до тех пор, пока не будут:

– деактивированы все вопросы, или

– выполнены все β - правила и не останется стартовых вопросов, или

– просмотрена таблица ChoiceRules. После того, как все её правила просмотрены, из таблицы Objects выбираются объекты, имеющие в поле _rowcount максимальное значение, т.е. которые были выбраны

наибольшее число раз.

8.2 Отличия функционирования ЭС гибридного и автономного классов

Помимо описанных структурных отличий таблиц однотипных правил, класс допустимых гибридных ЭС имеет и некоторые другие отличия:

1) Вместо изменения параметров путём ответа на вопрос (т.е. выполнения командного выражения поля param_values ответа из таблицы Answers), значения таких параметров считываются из указанного файла (естественно, что такие параметры должны существовать в таблице параметров ЭС). Формат данных, которые принимаются интерпретатором, описан в формуле (13) формализованной модели. Единицы таких данных – команды – непосредственно в файле должны быть написаны с новой строки (т.е. в конце каждой команды должны стоять символы перевода строки и возврата каретки). Команды считываются до тех пор, пока не будет достигнут конец файла. В таком случае машина вывода ожидает следующую команду в течение времени, заданного разработчиком данной модели ЭС, и, при её отсутствии в файле, вывод считается завершённым, и вызывается функция, сигнализирующая об этом.

2) Отсутствует таблица β - правил, и, следовательно, этап, соответствующий их просмотру;

3) Таблица γ - правил просматривается каждый раз после окончания цикла просмотров таблицы α - правил;

4) Вывод может быть досрочно остановлен. Способ остановки вывода зависит от выбранной модели вывода для гибридных ЭС. Остановка вывода подразумевает не вывод сообщения, а запуск сигнализирующей функции.

8.3. Алгоритм вывода для F-модели класса гибридных ЭС

F-модели ЭС данного класса предполагает использование δ - правил – правил завершения вывода. Каждое такое правило может быть активировано триггером, срабатывающим при обновлении полей таблицы Params, т.е. при изменении некоторых параметров. Данный триггер анализирует поле param_name всех правил таблицы FinalizationRules, и, в зависимости от их значений, устанавливает в 1 свойство активности правила.

Интерпретатор анализирует δ - правило только в том случае, если оно активно. Такой анализ производится сразу после анализа γ - правил

таблицы ChoiceRules, и любое правило, условное выражение (поле condition) которого верно, сигнализирует о прекращении процесса вывода досрочно. После этого вызывается сигнализирующая функция с параметрами, позволяющими получать выбранные ранее объекты из базы данных (с максимальным значением поля _rowcount).`

Этапы вывода для данной модели:

1) Считывание команды. При повторном неудачном считывании по прошествии тайм-аута вывод останавливается;
2) Циклический просмотр и выполнение α - правил. После выполнения некоторых α - правил может сработать триггер активации δ - правил;
3) Выполнение γ - правил;
4) Проверка условий активных δ - правил и завершение вывода в случае правильности хотя бы одного из них;
5) Возврат на 1 этап.

8.4. Алгоритм вывода для N-модели класса гибридных ЭС

Алгоритм вывода для данной модели отличается от алгоритма вывода F-модели тем, что вывод останавливается, как только число выбранных объектов после выполнения γ - правил не станет меньшим или равным числу N, задаваемому на этапе создания ЭС при выборе этой модели вывода.

Этапы вывода для данной модели:

1) Считывание команды. При повторном неудачном считывании по прошествии тайм-аута вывод останавливается;
2) Циклический просмотр и выполнение α - правил;
3) Выполнение γ - правил;
4) Проверка условия выборки объектов. Если число выбранных объектов не больше N, то вывод останавливается;
5) Возврат на 1 этап.

Пример функционирования рассмотренных моделей ЭС

Рассмотрим логику вывода в экспертной системе, разработанной на основе данной реляционной модели. В данном примере ЭС позволяет пользователю выбрать оптимальную модель зеркального плёночного фотоаппарата (используются упрощённые схемы данных).

В системе известны объекты выбора – ими являются фотоаппараты, имеющие атрибуты brand (фирма), model (модель),

min_shutter(минимальная скорость срабатывания затвора), cost (стоимость) и т.д.

Названия столбцов в данном случае представляют собой параметры селективного объекта. Поле _rowcount является системным, а ergonomic, functionality и class – атрибутами, оценивающимися субъективно.

1	1	Nikon	F55	2000	285	5	5	290	1	90	2	1	0
2	1	Nikon	F65	2000	310	5	5	330	1	90	2	2	0
3	1	Nikon	F75	2000	350	5	5	320	1	90	3	2	0
4	1	Nikon	F80	4000	450	1	5	515	1	125	2	3	0
6	1	Minolta	Dynax40	2000	230	5	7	260	1	90	2	1	0
8	1	Minolta	Dynax60	2000	330	5	9	310	1	90	2	3	0
9	1	Minolta	Dynax7	8000	650	5	NULL	575	0	200	2	3	0
10	1	Pentax	MZ-6	4000	260	5	3	385	0	125	2	3	0
11	1	Pentax	MZ-7	2000	270	5	3	380	0	100	2	2	0
14	1	Pentax	*ist	4000	400	5	3	335	1	NULL	3	1	0
15	1	Pentax	MZ-S	6000	1000	1	NULL	520	0	NULL	2	3	0
17	1	Sigma	SA-9	8000	310	1	1	435	1	NULL	1	3	0
18	1	Canon	EOS-300	2000	320	6	7	350	1	NULL	2	2	0
19	1	Canon	EOS-300V	2000	350	6	7	350	1	NULL	3	2	0
21	1	Canon	EOS-3000N	2000	270	5	7	350	1	NULL	2	2	0
22	1	Canon	EOS-33	4000	450	8	7	575	0	NULL	2	3	0

Пусть система находится на стадии обработки ответов на вопросы. Таблица параметров в данный момент времени содержит следующие значения:

key_params	user_iq	user_yearpro	max_cost	have_lenses	macro_req	brand	centerweight	design	color	user_plans	programs	kit	other_brand
90	75	0	400	no	1	NULL	0	e	silver	art+objects	1	1	1
91	90	0	400	NULL	NULL	NULL	NULL	f	anything	NULL	NULL	NULL	0

Параметры, которые могут быть использованы при выборе фотоаппарата: user_iq (уровень знаний пользователя), macro_req (требуется ли макро-режим), brand (фирма) и т.д.

Серым в таблице выделены параметры предыдущей сессии моделирования. Для текущей сессии используется параметрический объект с уникальным для всех сессий моделирования идентификатором

key_params = 91 (так, для следующей сессии будет создана строка параметров и номером, большим на 1 (свойство IDENTITY) - 92).

Пусть пользователю задан вопрос (question), предполагающий ответ типа R (answer_type) – несколько заранее определённых вариантов ответа. Этот вопрос в настоящий момент активен (active = 1), поэтому может быть задан.

Допустим, пользователь ответил на ответ с номером 73 (или, в случае гибридной ЭС, была получена команда «user_yearpro=0»). Тогда сбрасывается активность данного вопроса, и в таблице параметров (параметрический объект) параметр с названием «user_yearpro» делается равным 0 (нулю):

```
UPDATE params SET user_yearpro=0
WHERE key_params = 91
```

Состояние системы меняется:

key_params	user_iq	user_yearpro	max_cost	have_lenses	macro_req	brand	centerweight	design	color	user_plans	programs	kit	other_brand
91	90	0	400	NULL	NULL	NULL	NULL	f	anything	NULL	NULL	NULL	0

Просматриваем таблицу α - правил. Участвуют только активные правила:

key_rule	active	condition	param_values	inactive_questions
9	1	(user_yearpro=0)	programs=1, have_lenses='no, kit=1	3,10,24,17,11
2	1	(user_plans='home')	macro_req=0	5
18	1	(have_lenses='no')	other_brand=1	17,24
15	1	(user_iq<25) AND (user_plans='home')	centerweight=0	18

Запускаем все правила на выполнение с помощью запроса и де-

активируем соответствующие вопросы:

```
UPDATE Params SET <param_values>
   WHERE <condition> AND key_params = <n>
go
UPDATE Questions SET active=0
   WHERE active <> 0 AND <inactive_questions>
```

При этом выполнится только 9-ое правило, т.к. изменился только параметр user_yearpro (предикат WHERE будет верным только в данном случае):

```
UPDATE params
SET
      programs=1, have_lenses='no', kit=1
WHERE
      (user_yearpro=0) AND key_params = 91
```

и деактивируем список вопросов:

```
UPDATE Questions
SET active = 0
WHERE active <> 0 AND key_question in
(3,10,24,17,11)
```

Получим следующее состояние системы:

key_params	user_iq	user_yearpro	max_cost	have_lenses	macro_req	brand	centerweight	design	color	user_plans	programs	kit	other_brand
91	90	0	400	no	NULL	NULL	NULL	f	anything	NULL	1	1	0

Просматриваем условные правила, пока меняется состояние системы. В нашем случае, после выполнения 9-го правило, это состояние поменялось. Поэтому заново просматриваем таблицу α - правил. Так как параметр have_lenses стал равным "no", то должно выполниться 18 правило и снова изменить состояние системы:

key_params	user_iq	user_yearpro	max_cost	have_lenses	macro_req	brand	centerweight	design	color	user_plans	programs	kit	other_brand
91	90	0	400	no	NULL	NULL	NULL	f	anything	NULL	1	1	1

и так далее.

Теперь необходимо выбрать следующий вопрос (только в случае автономной ЭС), таблица JmpRules (β - правила):

key_rule	active	condition	next_question_id
3	1	(user_yearpro!=0)	3
8	1	(have_lenses='h')	24
9	1	(have_lenses NOT IN ('h','no'))	17
12	1	(have_lenses!='no') AND (other_brand=1)	6

Запоминаем все номера вопросов, которые могут быть заданы следующими в случае их активности:

```
SELECT next_question_id
   WHERE active = 1 AND <condition>
```

Задаём следующий вопрос из полученного списка их идентификаторов next_question_id и деактивируем те β - правила, которые выбрали данный вопрос (выполнилось условие в предикате предыдущего запроса). Важным является то, что если вышеобозначенный список «заканчивается», и на некоторой итерации уже не из чего выбирать, но ещё есть активные вопросы, то задаётся тот, который может быть стартовым (т.е. может быть задан вне зависимости от того, были ли заданы какие-либо другие вопросы) - это определяется по полю class, которое равно 0 для независимого (стартового) вопроса. Если же таковых нет, то вывод переходит на стадию выполнения γ – правил.

Рассмотрим следующий набор γ – правил (таблица ChoiceRules):

key_choice_rule	basic	condition	param_name	param_values_if
8	2	NOT(kit IS NULL)	kit	kit=*
38	13	other_brand=1 AND NOT (brand IS NULL)	brand	brand='*'
12	2	programs=1	NULL	programs>=5
7	0	NULL	max_cost	*>=cost
35	15	programs=0	NULL	programs<5

Столбец param_name содержит название атрибута параметрического объекта (характеристики фотоаппарата), а param_values_if – условие для выбора селективного объекта.

Сначала необходимо просмотреть всю таблицу данных правил и

проверить истинность условия поля condition для каждого правила:

```
EXISTS
      (SELECT * FROM Params
      WHERE <condition> AND key_params = <n>),
```

После проверки условий и сортировки правил по приоритетам (basic) получаются следующие результаты (серым помечены не прошедшие проверку правила - см. предыдущую таблицу параметров):

key_choice_rule	basic	condition	param_name	param_values_if
7	0	NULL	max_cost	*>=cost
8	2	NOT(kit IS NULL)	kit	kit=*
12	2	programs=1	NULL	programs>=5
38	13	other_brand=1 AND NOT (brand IS NULL)	brand	brand='*'
35	15	programs=0	NULL	programs<5

Теперь необходимо сформировать условие, на основе которого будет выбран непосредственно объект. Для этого для каждой группы правил с одинаковым приоритетом формируется условие из поля (param_values_if), где (*) заменяется на имя параметра, а условия объединяются в одно, если количество правил с текущим приоритетом больше одного:

```
SELECT key_object FROM Objects
WHERE <condition> AND key_object IN <objects>
```

или для γ - правил - 7, 8, 12, 38

```
--- правило 7
первого
SELECT key_object FROM Objects
WHERE
   (
            (SELECT max_cost FROM Params
            WHERE key_params = 91) >=cost
      )
   AND key_object IN <objects n>
```

```
UPDATE Objects SET _rowcount = _rowcount+1
WHERE key_object IN <objects n>
go


--- правила 8 и 12 с одинаковым приоритетом
----- n = n+1

SELECT key_object FROM Objects
WHERE
     (
          (SELECT kit FROM Params
          WHERE NOT(kit IS NULL)
          AND key_params = 91) =kit
   OR
          programs>=5
   )
   AND key_object IN <objects n-1>

UPDATE Objects SET _rowcount = _rowcount+1
WHERE key_object IN <objects n>
go

----- n = n+1

SELECT key_object FROM Objects
WHERE
   (
           (SELECT brand FROM Params
           WHERE other_brand=1 AND NOT (brand IS
      NULL)
           AND key_params = 91) =brand
   )
   AND key_object IN <objects n-1>

UPDATE Objects SET _rowcount = _rowcount+1
WHERE key_object IN <objects n>
go
```

Каждый запрос при этом возвращает массив идентификаторов объектов <objects n>, который используется при выборе объектов для следующей группы правил с одинаковым приоритетом (<objects n> содержит идентификаторы всех объектов только для первого выполняющегося правила, далее количество элементов массива не увеличивается).

Следующий за запросом оператор UPDATE увеличивает поле _rowcount для каждого выбранного на данном шаге объекта. Таким образом, когда все правила будут выполнены, наиболее подходящими станут те объекты, у которых поле _rowcount имеет максимальное значение.

В случае если используется N-модель вывода для ЭС гибридного типа, то, как только будет найдено установленное пользователем число подходящих объектов, вывод будет остановлен (см. п. 8.2); об использовании F-модели см. п. 8.3.

Особенности применения рассмотренного подхода. Достоинства и недостатки.

Как было показано выше, использование реляционного подхода при проектировании экспертных систем может стать подходящим решением для разработки ЭС как гибридного, так и автономного классов. Использование реляционных отношений и физических структур позволяет в значительной степени уменьшить трудоёмкость разработки системы, увеличить интеграционные характеристики и снизить стоимость её реализации, и, в конечном счёте, стоимость внедрения.

Подводя общие итоги и анализируя возможности реляционных систем, следует выделить некоторые особенности реализации экспертных систем на основе СУБД. К достоинствам их реализации относятся следующие аспекты:

1) Такие системы могут быть распределёнными. При этом всё управление распределенными запросами ложится на СУБД и происходит скрыто от пользователей. Программирование распределённых систем, с точки зрения T-SQL, не является сложной операцией. Помимо распределённых запросов, возможно сегментировать базы знаний, представленные в форме БД, получив таким образом уже иной класс систем.

2) Использование средств реляционной СУБД позволяет ПОЛНОСТЬЮ перенести реализацию машины вывода внутрь такой системы. Соответственно, изменение серверной логики не затронет клиентов, получающих информацию от ЭС (в случае использования клиент-серверной архитектуры типа "толстый клиент", инженерам-программистам интеллектуальной системы, при необходимости внесения изменений в код интерпретатора, придётся заново распространять новую версию клиентского ПО).

3) Стандартные средства СУБД включают эффективный синтаксический анализатор. Набор встроенных операторов и функций позволяет реализовать достаточно сложную логику вывода или легко исполь-

зовать "пустые" запросы для проверки некоторого выражения. Более того, в настоящее время СУБД позволяют использовать, помимо стандартного T-SQL, внешние языки программирования высшего уровня, - следовательно, возможна более эффективная и удобная реализация функциональности интерпретатора.

4) Специальные функции СУБД, как было продемонстрировано в примере, позволяют создавать также событийно – ориентированные модели вывода: представленные δ-правила позволили эффективно использовать возможности триггеров T-SQL и реализовать стратегию фокусирования.

Ограничения и недостатки, которыми обладают реляционные СУБД, являющиеся базой для разрабатываемых систем экспертного моделирования:

1) Как уже было отмечено ранее, обладая рассмотренной структурой, схема отношения на множестве правил вывода не может быть чётко определена. Поэтому, используя функциональность синтаксического анализатора СУБД, приходится жертвовать тем, что при количестве правил, превышающих несколько десятков, отследить связи между ними и используемыми в них параметрами становится проблематично. В таком случае появляется необходимость в реализации дополнительного модуля внесения и редактирования знаний.

2) Несмотря на отсутствие ограничения на количество строк в таблицах БД, существует ограничение на количество их столбцов и размерность типов данных. Это означает, что реализовать объект с большим количеством атрибутов невозможно. Данное ограничение не является таковым для подавляющего большинства (если не для всех) предметных областей; однако с ростом сложности и, соответственно, длины текстовых полей, представляющих собой посылку или условие некоторого правила, может возникнуть необходимость использования дополнительных средств расширения размерности, что может повлиять на производительность запросов или сделать невозможным их выполнение.

3) Весомым недостатком по сравнению со специализированными системами моделирования, где используются специальные языки представления знаний, операторы и функциональность, является то, что СУБД не позволяют настолько же эффективно (с точки зрения производительности) осуществлять вывод и использовать ресурсы, так как они создавались для решения совершенно других задач. По этой причине невозможно создание полноценной и производительной гибридной динамической (а в некоторых случаях и квазидинамической) экспертной

системы. Особенно удачным решением будет реализация автономных экспертных систем, так как они менее требовательны к характеристикам производительности.

Литература

1. Курс лекций по реляционной алгебре [электронный ресурс] / МИФИ

2. Методология общесистемного проектирования [рукопись] : курс лекций / Григорьев Ю.А. - 2006 г. – МГТУ

3. Интеграция систем ситуационного, имитационного и экспертного моделирования [электронный ресурс] : (ИИ, ситуационные центры, моделирование, полиграфия) / Филиппович А.Ю. – М.: Изд-во "ООО Эликс+". - 2003. – 300 с. – Электрон. текст. данн. – Режим доступа: http://www.philippovich.ru/Library/Articles/publications_Philippovich_Andrew/ mypublicationsinet/mypublicationsinet.HTM

4. Интеллектуальные системы [рукопись] : курс лекций / Филиппович А.Ю. – 2005 г. - МГТУ

5. Моделирование деятельности, профессиональное обучение и отбор операторов [электронный ресурс] / под ред. Е.К. Масловского. – М.: Мир. – 1991. - гл. 2: Искусственный интеллект / Кинг Сунь Фу. – (Человеческий фактор : в 6 т. / под ред. Г. Салвенди; т. 3). - Электрон. текст. дан. - Режим доступа: http://www.librus.ru

6. Интеллектуальные системы [электронный ресурс] : Эл. публикация / Сибирский государственный университета управления. – Электрон. текст. данные. – Режим доступа: http://www.ssti.ru/kpi/informatika/Content/biblio/b1/inform_man/gl16x.htm

7. Рынок СУБД: show must go on! [электронный ресурс] : Аналитика / Николай Печерица.- Электрон. текст. дан. – Режим доступа: http://www.silicontaiga.ru

Т.В.Колобкова

МАКЕТ ЭЛЕКТРОННОГО ИЗДАНИЯ МУЗЕЙНОГО СОБРАНИЯ РУКОПИСЕЙ

Введение

За последние двадцать-тридцать лет значительно оживился интерес к истории материальной и духовной культуры древней и средневековой Руси. Интерес этот наблюдается не только у нас в стране, но и далеко за ее пределами.

Широкие круги ученых-обществоведов — археологи, историки, литературоведы, фольклористы, искусствоведы, правоведы, и конечно, языковеды все чаще и чаще обращаются к древнерусским письменным памятникам как к первоисточникам познания многовековой, богатой политической и духовной жизни русского народа.

«Музейное собрание рукописей» Российской Государственной Библиотеки для исследователей представляет собой исчерпывающий по своей полноте справочник. В частности 2-ой том собрания включает рукописные материалы, поступившие в 1862-1920 гг. (№№ 3006-4500). Согласно Описи 1968 года, 2-ой том насчитывает 1494 номеров. В число их входят единицы хранения, в дальнейшем присоединенные к самостоятельным фондам (для рукописей, выделенных в отдельные рукописные фонды, имеются подробные библиографические справки с указанием имеющихся печатных и рукописных описаний этих фондов).

Наиболее важные достоинства Музейного собрания рукописей – это, прежде всего, полнота и тщательность самих описаний, дающих постатейную роспись содержания сборников, сведения о редакциях и иных разновидностях текстов представленного в собрании памятника, соотнесения с существующими его публикациями и исследованиями, а также такие особенности, как хронологический принцип расположения описываемых рукописей (в последовательности их поступления) и нумерация статей при описании сборников, что весьма существенно для полноценности справочного аппарата книги.

Важной чертой являются, конечно, и строго выдержанная полнота формальной характеристики каждой рукописи (её почерк, количество листов, формат, переплёт, наличие украшений или нотировок, фиксация всех владельческих записей и т.д.), аргументированность датировок по

водяным знакам бумаги с точным отсылками к номерам соответствующих альбомов филиграней.

В жанровом отношении Музейное собрание исключительно разнообразно. Исторические, литературные, богослужебные памятники древнерусской письменности, произведения русской публицистики, философские сочинения, переводная литература, эпистолярное наследие — все это включает в себя 2-ой том Музейного собрания.

1. Разработка макета электронного издания

Электронное издание представляет собой совокупность графической, текстовой, цифровой, речевой, музыкальной, видео -, фото - и другой информации. В одном электронном издании могут быть выделены информационные (или информационно-справочные) источники, инструменты создания и обработки информации, управляющие структуры. Электронное издание может быть исполнено на любом электронном носителе, а также опубликовано в информационной компьютерной сети.

Разработка макета электронного издания включает в себя следующие этапы: подготовка материалов книги и создание макета электронного издания.

Подготовка материалов книги подразумевает ввод текста и иллюстраций в компьютер. Для этого разработана технология ввода информации.

Макет разрабатывается в среде Macromedia Director. Macromedia Director – первоклассный творческий инструмент, позволяющий создавать приложения с большим набором средств – графикой с высоким разрешением, анимацией, звуковыми эффектами и цифровыми фильмами. Director является одним из самых используемых мультимедийных инструментов для решения любых задач – от презентаций до полнофункциональных заголовков для CD и компьютерных игр.

2. Технология ввода информации

Схема технологического процесса представлена на рис. 1. Этапы, процедуры и операции ввода книги представлены в табл. 1.

Таблица 1. Основные операции ввода книги.

Этапы	Процедуры	Операции
1. Автоматизированный ввод текстовой информации	1.1. Установка параметров сканирования, сегментации, распознавания и	1.1.1. Настройка параметров сканирования 1.1.2. Настройка параметров сегментации

	проверки	1.1.3. Настройка параметров распознавания 1.1.4. Настройка параметров проверки орфографии
	1.2. Сканирование книги	1.2.1. Пробное сканирование 1.2.2. Сканирование всей книги
	1.3. Редактирование отсканированных страниц	1.3.1. Проверка ориентации изображении и их исправление 1.3.2. Очистка изображения от «загрязнения» в виде черных точек
	1.4. Сегментация	
	1.5. Редактирование отсегментированных страниц	1.5.1. Проверка размера блоков 1.5.2. Проверка типа блоков
2. Распознавание отсканированных изображений	2.1. Распознавание	
3. Внесение корректуры в распознанный текст	3.1. Редактирование распознанного текста и проверка орфографии	
4. Создание документа MS Word	4.1. Создание документа	4.1.1.Создание нового документа 4.1.2. Настройка параметров его страниц 4.1.3. Сохранение в файл
	4.2. Экспорт текста в MS Word	
	4.3. Редактирование документа в MS Word	4.3.1. Правка текста книги. 4.3.4. Создание колонтитулов 4.3.5. Вставка страниц

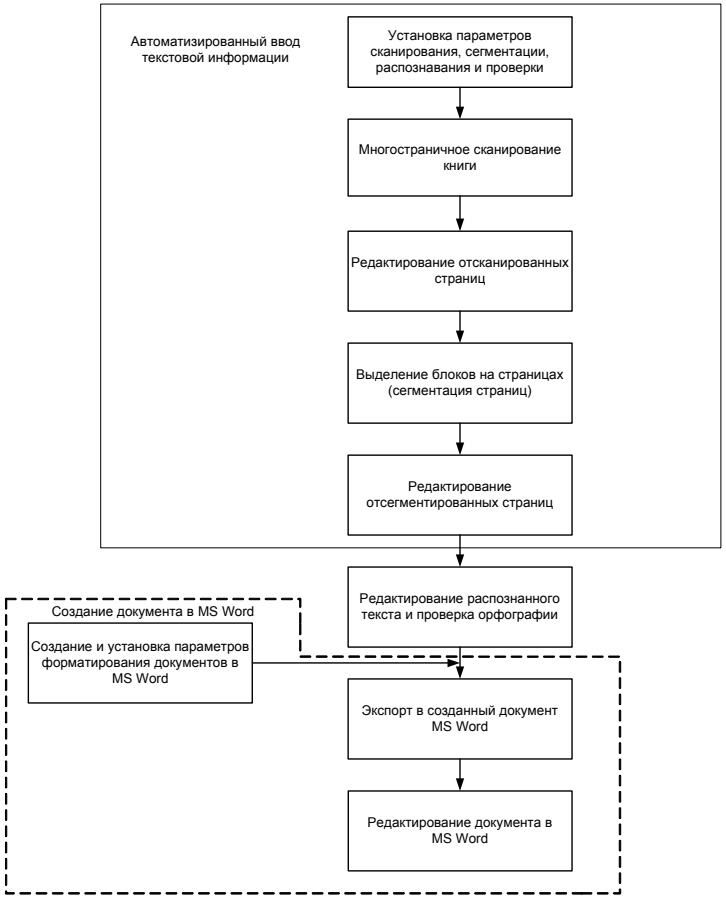


Рис. 1. Схема технологического процесса ввода текста книги.

Установка параметров сканирования, сегментации, распознавания и проверки

а) Операция настройки параметров сканирования.

Сканирование производится с помощью программного обеспечения (ПО) FineReader фирмы ABBYY и сканера “HPII Scan” корпорации Hewlett Packard. Главными параметрами сканирования являются: разрешение получаемого изображения (resolution); контрастность (contrast); яркость (brightness).

Яркость: для светлых документов яркость необходимо установить ниже (сделать их темнее), для темных - выше (сделать их светлее).

Разрешение: 300 dpi - для большинства документов; 400 - 600 dpi - для документов, набранных мелким шрифтом.

Контрастность: в данном случае оставляем по умолчанию.

б) Операция настройки параметров сегментации.

Сегментация происходят быстрее, если обрабатываемое изображение не показывается на экране компьютера. Чтобы автоматизировать процедуру сегментации, необходимо:

1. Из меню «Сервис» выбрать пункт «Опции…»
2. В диалоге «Опции» выбрать закладку «Сегментирование».
3. В группе «Расположение текста» выбрать пункт «Автоматическое определение».

в) Операция настройки параметров распознавания.

Главными параметрами распознавания являются язык распознавания и тип текста.

При распознавании текста книги необходимо выбрать нужный язык, слова которых встречаются в распознаваемом тексте, из списка на панели Распознавание. Тип текста определяется в системе автоматически.

г) Операция настройки параметров проверки орфографии.

При проверке орфографии есть следующие возможности: останавливаться на правильных неуверенно распознанных словах, на редких формах, между страницами при проверке орфографии и автоматически корректировать пробелы до и после знаков препинания.

Операция сканирования книги

Операция пробного сканирования.

Пробное сканирование необходимо для того, чтобы определить качество сканирования страницы книги. После того, как отсканирована страница, можно настроить оптимальные параметры сканирования и установить границы сканирования. Результаты настройки обычно сразу отображаются в окне предварительного просмотра изображения. По полученному результату мы определяем границы (участок) сканирования, как показано на рис. 2, а также регулируем яркость.

Операция редактирования изображения

а) Проверка ориентации изображении и их исправление.

Распознаваемое изображение должно иметь стандартную ориентацию: текст должен читаться сверху вниз и строки должны быть парал-

лельны нижнему краю экрана. Если изображение было отсканировано в вертикальном виде, то его необходимо перевести в стандартный вид.

Рис. 2. Предварительный просмотр

б) Очистка изображения.

Из-за плохого качества бумаги на отсканированных изображениях возможен «мусор» в виде черных точек, поэтому необходимо очистить их, чтобы ПО лучше распознавала изображения.

Операция сегментации

Для того, чтобы автоматически сегментировать все изображения, из меню «Scan&Read» выберите пункт «Сегментировать все страницы...». Результат этой операции: изображения покрылись прямоугольниками. Номера на прямоугольниках означают порядковый номер процесса распознавания, а цвет границы означает тип блока.

По умолчанию ПО FineReader назначает цвета: серый цвет – для нераспознаваемых символов, слов (формулы); зелёный цвет – текстовый блок; коричневый цвет – блок “таблица”; красный цвет – блок “рисунок”; ярко-зеленый цвет – блок “штрих-код”.

В левой части ПО значок изображения изменился на зеленоватый цвет (рис. 3).

Редактирование отсегментированных страниц

В процессе редактирования можно добавлять, удалять и видоизменять блоки с помощью контекстного меню (выделить блок правой кнопкой мыши – рис. 4). Один из некоторых случаев исправления бло-

ков показан на рис. 4. После того, как все страницы правильно выделены блоками, можно приступать к распознаванию текста.

Рис. 3

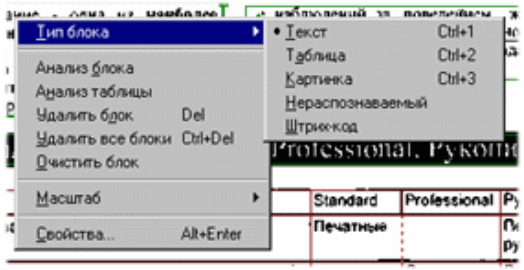


Рис.4. Контекстное меню при выделении блока

Операция распознавания текста

Процесс распознавания текста – получение текста из изображения, ключевой момент в технологии ввода текста. Процесс распознавания происходит автоматически без участия пользователя путем выбора пункта «Распознать все нераспознанные страницы» в меню «Scan&Read». Признаком окончания процесса распознавания является то, что в середине экрана вместо графического изображения появился текст, и в левой части значок изменился.

Операция редактирования и проверки орфографии

Проверка орфографии начинается с выбором пункта «Проверить» в меню «Сервис». Работа в диалоговом окне «Проверка орфографии». Чтобы вернуться к документу после того, как проверка закончена, нажать кнопку «Закрыть» в диалоге «Проверка орфографии».

Создание документа MS Word

Настройка параметры страниц.

Выбрать пункт «Параметры страницы…» в меню «Файл», после чего появится диалог «Параметры страницы».

В нашем случае необходимо выбрать следующие настройки:

1. Закладка «Макет документа»: размер бумаги – А4; ориентация – книжная; применить – ко всему документу.

2. Закладка «Макет»: различать колонтитулы – четных и нечетных страниц, первой страницы; применить – ко всему документу.

3. Закладка «Поля»: отметить «Зеркальные поля»; верхнее – 2,54; нижнее – 1,27; левое – 2,54; правое – 5,04; от края до колонтитула – верхнего 1,27 и нижнего 1,27; применить – ко всему документу.

Эффективность технологического процесса ввода текстовой информации

Этап 1. Автоматизированный ввод текстовой и графической информации	$T_{установки} = T_{открытия\ ПО} + T_{вызов\ окна\ настройки} + T_{настройки}$	2,65 ч.
	$T_{сканиров.общее} = (T_{подготовки} + T_{сканирования}) *$ кол-во сканир. Стр.	
	$T_{редакт.\ изобр.} = (T_{поворота} + T_{очистки}) *$ кол-во отсканир. Стр. $+ T_{сохр.пакета,}$	
	$T_{сегментация\ общ.} = (T_{выделения\ блоков} + T_{проверка\ блоков}) *$ кол-во отсканиров.стр.	
Этап 2. Распознавание.	$T_{этап\ 2} = T_{распозн..стр} *$ кол-во отсканир.стр.	1,125ч.
Этап 3. Внесение корректуры	$T_{этап\ 3} =$ среднее время проверки* кол-во отсканир.стр.	45 ч.
Этап 4. Создание документа MS Word	$T_{создания} = T_{открытия\ ПО} + T_{вызов\ окна\ настройки} + T_{настройки} + T_{сохранения}$	25,63ч.
	$T_{экспорт}$	
	$T_{редакт.} = T_{создания\ колонтитула} + T_{вставки\ страниц} + T_{правки\ текста}$	
	$T_{создания\ колонтитула}$ включает время создания колонтитула = мин	
	$T_{правки\ текста} =$ ср. время правки* кол-во отсканир.стр.	
Итого:	$T_{ввод} = T_{этап\ 1} + T_{этап\ 2} + T_{этап\ 3} + T_{этап\ 4}$	74,4 ч.

Ввод книги составляет приблизительно 75 часов, что составляет в среднем 50 минут на 1 страницу книги. Эта величина является нормальной ввиду специфики текста и лексики.

Процесс ввода специфической текстовой информации – очень тяжёлая и трудная работа с точки зрения временных затрат. Большое количество времени уделяется на корректуру текста, но ввод текста очень сильно зависит от результатов сканирования, например, от качества сканируемого материала и от настройки параметров сканирования.

На основании результатов можно сказать, что ввод технической текстовой информации сложен и требует очень много ресурсных затрат, чем ручной ввод текстовой информации.

3. Создание макета электронного издания

Создание макета включает следующие этапы:

1. Разбивка текста по страницам виртуальной книги.
2. Реализация ссылок перелистывания страниц виртуальной книги.
3. Реализация ссылок перехода из списка иллюстраций и из содержания к соответствующей странице.
4. Реализация ссылок переходов к Содержанию, Списку иллюстраций и Выход.
5. Осуществление увеличения изображений.

Разбивка текста по страницам виртуальной книги

Страница виртуальной книги меньше по размерам страницы реальной книги, поэтому необходимо разбить исходный текст и равномерно распределить. Причем для наглядности удобно иллюстрацию разместить слева, а справа – текст к этой иллюстрации. Также необходимо, сохраняя структуру книги, разместить вверху страниц соответствующие заголовки (Поступления 1887 года, Список иллюстрации и т.д.).

Реализация ссылок перелистывания страниц виртуальной книги

Для перелистывания страниц виртуальной книги используются ссылки на каждой странице книги в виде стрелок (см. рис.5). Для реализации перехода по страницам необходимо написать небольшой скрипт. Для перехода к следующей страницы:

```
on mouseUp
          go next
          sprite(8).cursor=-1
end
```

Причем для наглядности необходимо осуществить некоторое подсвечивание стрелок и появление подсказки («Следующая страница» или «Предыдущая страница») при наведении на них курсора мышки.

Реализация ссылок перехода из списка иллюстраций и из содержания к соответствующей странице

При наведении курсора мышки на пункт содержания или пункт списка иллюстраций, он выделятся более жирным шрифтом, как показано на рис.5 (выделен пункт 14). А при нажатии кнопки мыши осуществляется переход на соответствующую страницу виртуальной книги.

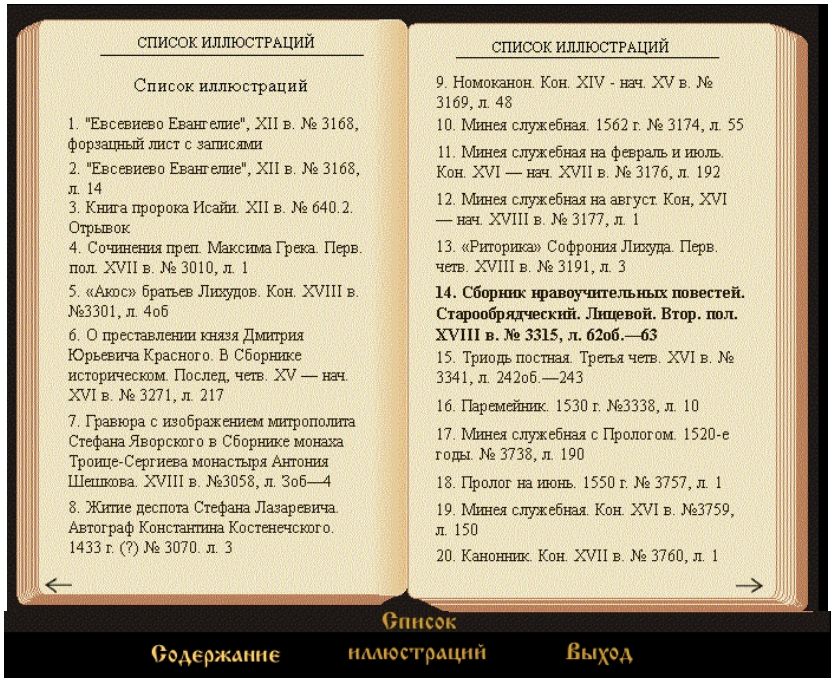


Рис.5. Список иллюстраций виртуальной книги

Для реализации этого необходимо написать скрипт:

```
property UpPosition
Property highlite
property pGoToMarker

on mouseEnter me
```

```
    set the member of sprite the spritenum of me =
member highlite
  end
  on mouseLeave me
    set the member of sprite the spritenum of me =
member upPosition
  end
  on mouseUpOutside me
    set the member of sprite the spritenum of me =
member upPosition
  end
  on mouseUp me
    set the member of sprite the spritenum of me =
member upPosition
    go frame pGoToMarker
  end
  on getPropertyDescriptionList me
    theMember = the member of sprite the cur-
rentspritenum
      set p_list = [:]
    addprop p_list, #UpPosition, [#comment: "Up
Button:", #format: #member, #default: theMember ]
    addprop p_list, #DownPosition, [#comment: "Down
Button:", #format: #member, #default: theMember]
    addprop p_list, #highLite, [#comment: "Highlite
Button:", #format: #member, #default: theMember]
    addprop p_list, #pGoToMarker, [#comment: "GoTo
Marker:", #format: #marker, #default: 1]
    return p_list
  end
```

В специальном окне задаются параметры:

Up Button – вид ссылки (в нашем случае стрелки) при не нажатом состоянии;

Down Button – вид ссылки после нажатия на неё мышкой

Highlite Button – вид ссылки (т.е. подсвечивание) при наведении на неё курсора мышки.

GoTo Marker задает номер виртуальной страницы, к которой осуществляется переход при нажатии на ссылку мышкой.

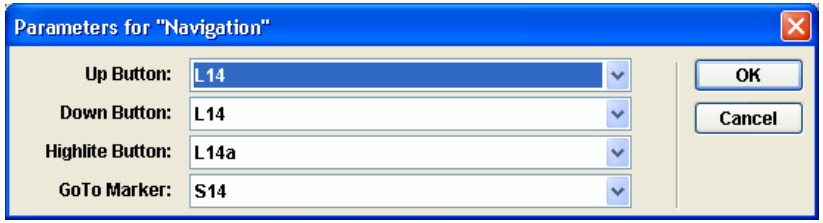


Рис.6. Параметры реализации ссылок перехода из Списка иллюстраций.

Реализация ссылок переходов к Содержанию, Списку иллюстраций и Выход

Для быстрого перехода из любой страницы виртуальной книги к содержанию и списку иллюстраций, а также для выхода из приложения реализованы три ссылки внизу страницы (рис.5), соответственно: «Содержание», «Список иллюстраций» и «Выход». При наведении на них курсором мышки также осуществляется подсвечивание ссылок, а при нажатии на ссылку соответствующее действие: либо переход к необходимой странице виртуальной книги, либо выход из приложения.

Для реализации этих переходов используется скрипт, с соответствующими параметрами, описанный в пункте 5.3.

Осуществление увеличения изображений

Увеличение изображения осуществляется при нажатии на них кнопкой мышки. При этом при попадании курсора мышки в поле иллюстрации курсор меняет свое изображение, т.е. из стрелки меняется на изображение  или  . Если иллюстрация в исходном состоянии, то при наведении на неё курсор мыши меняется на  , далее если нажать кнопку мышки, то появится увеличенное изображение иллюстрации и курсор мыши меняется на  (рис.7). Если нажать опять кнопку мыши, то иллюстрация примет исходное состояние.

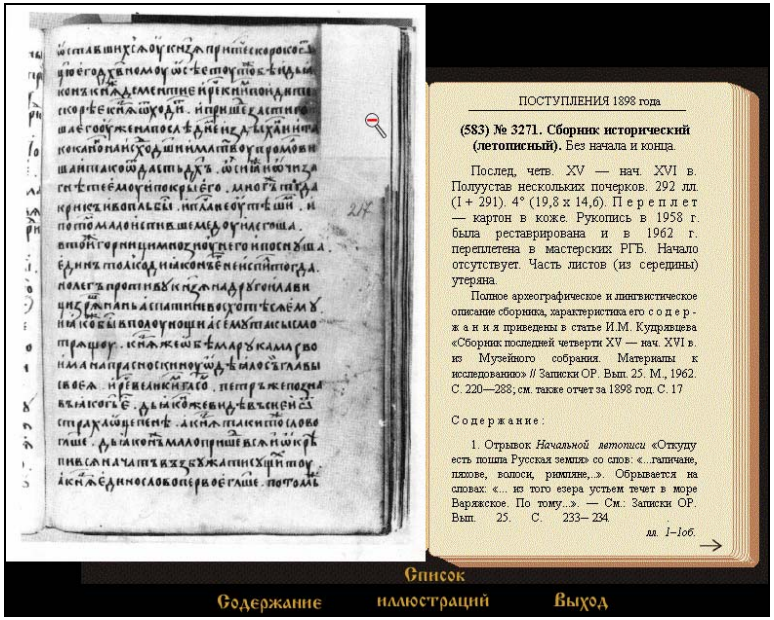


Рис.7. Увеличенное изображение

Заключение

В заключении важно отметить, что представление информации в электронном виде дает несравнимо большие возможности для работы с данными, в отличие от традиционных изданий в печатном виде. Это позволяет обеспечить доступ к книгам, выдача которых по каким-либо причинам ограничена или невозможна, либо количество или качество которых в традиционных фондах не соответствует уровню требований читателей.

Литература

1. Музейное собрание рукописей. Описание. Том 2 (№ 3006 — № 4500). М.: Научно-издательский центр «Скрипторий», 1997. — 496 с.: илл.
2. *Ричард Сальватьера, Пол Вивер, Скотт Дж. Вилсон* DIRECTOR MX [руководство от macromedia]/Пер. с англ. — М.: КУДИЦ-ОБРАЗ, 2004. — 480 с.
3. *Окрасса Уоррен* Director 8.5/MX. Shockwave Studio: Пер. с англ. — М.: ДМК Пресс, 2003. — 432 с.: ил. (Самоучитель).

Ле Ба Хонг Минь

ИНФОРМАЦИОННАЯ ТЕХНОЛОГИЯ СОЗДАНИЯ ЛИНГВИСТИЧЕСКИХ БАЗ ЗНАНИЙ РУССКО-ВЬЕТНАМСКИХ СИСТЕМ АВТОМАТИЧЕСКОГО ПЕРЕВОДА

ВВЕДЕНИЕ

Актуальность работы. В этом мире, каждая страна - личная культура, и каждая культура представляет через естественный язык, который является инструментом человека, используется для представления знания и общения вместе.

В случае человека прибыли от различных стран, они используют различные языки, общение между ними в обычном генерале труден, таким образом для решения этого проблема, часто есть помощь переводчиков или кого ни бут изучает язык другого.

В настоящем время с непрерывным развитием экономики, науки и техники мира, требование общения между странами слишком быстро увеличивается и в некотором случае не навсегда есть эффективная помощь от переводчика и бумажных словарей. Поэтому возникновение системы машинного перевода или набора, современные электронные словари являются обязательными для помощи пользователей, использует иностранные языки более быстро и более эффективно. В этом время технология информатики быстро развивает, персональный компьютер уже стает сильной системой с высокой скоростей процессора и объемом памяти большой, поэтому необходимое техническое условие для создания системы машинного перевода в общее уже готов.

До тех пор, кроме бумажного Русско-Вьетнамского и Вьетнамско-Русского словари у нас еще нет систем электронного словари и Русско-Вьетнамского машинного перевода, поэтому в этом время мы можем начинать изучать и создать такие системы для удовлетворения потребности общения между нами.

Цель работы. Исследование существующих систем машинного перевода. Анализ технологий строительства систем машинного перевода. Развитие некоторые структуры словарей базы данных, которые служат в

системе МП и развития удобной окружающей среды работы с этими словарями.

Методы исследований. В качестве метода исследования был выбран сравнительный анализ программных продуктов и разработок, предлагаемых различных фирмами, исследователями в области машинного перевода. Изучение общего свойства, характеристики естественного языка, использование современного языка программирования и сильного СУБД для строительства системы БД и удобного интерфейса.

Научная новизна. До тех пор еще нет никаких систем Русско-Вьетнамского машинного перевода и еще нет стандартного электронного словари данной системой, поэтому новые следствия работы не только понятия о машинном переводе, и еще раз мы можем изучить метод анализ нефлективного языка похоже как Вьетнамский язык.

Практическая ценность. Разработанная архитектура могла стать основанием для строительства варианта исследования машинного перевода и набор словари, которые планируются использовать не только анализ синтаксического и еще анализ семантического Русского и Вьетнамского языка.

Содержание работы. Диссертация состоит из введения, шест разделов, заключения, списка литературы из ... наименований и приложений. Работа содержит ... страницу основного текста и 9 листы.

Во введении обоснована актуальность темы диссертационной работы, определены ее цель, научная новизна и практическая ценность

В первой главе приведён обзор и анализ существующих технологий строительства систем машинного перевода. Рассмотрены различные классификации систем машинного перевода. Так же приведена краткая история создания систем перевода.

Во второй главе приведен аналитический обзор систем МП, а также изложены технологии машинного перевода. Рассмотрены основные этапы машинного перевода.

В третьей главе приведен анализ морфологического, синтаксического и семантического русского языка и от этого построение структуры систем словари, которые служат в машинном переводе.

В четвертой главе Организационно-экономическая часть

В пятой главе Защита информации

В заключении формулируются основные выводы по результатам исследования.

Технологии создания систем машинного перевода

История машинного перевода

Впервые идея относительно возможности машинного перевода высказал Чарльз Бэббидж (1791-1871), разработавший в 1836-1848 гг. проект цифровой аналитической машины - механического прототипа электронных цифровых вычислительных машин, появившихся через 100 лет. Идея Ч. Бэббиджа состояла, что память объемом 1000 50-разрядных десятичных чисел (по 50 зубчатых колес в каждом регистре) можно использовать для хранения словарей. Ч. Бэббидж привел эту идею в качестве обоснования для запроса у английского правительства средств, необходимых для физического воплощения аналитической машины, которую ему так и не удалось построить

Фактически история машинного перевода начинается с эксперимента Джорджтаунского. Первая публичная демонстрация машинного перевода с русского языка на английский, осуществленного на машине ИБМ-701 в январе 1954 г. Сообщение об этом событии было опубликовано в журнале Computers and Automation, 1954, № 2. А реферат этого сообщения, сделанный Д. Ю. Пановым, появился в РЖ ВИНИТИ "Математика", 1954, № 10: "Перевод с одного языка на другой при помощи машины: отчет о первом успешном испытании".

Это сообщение явилось толчком для начала работ по машинному переводу в СССР. Д. Ю. Панов, бывший тогда директором ВИНИТИ (в то время Института научной информации - ИНИ) привлек к работам по машинному переводу И. К. Бельскую, которая затем возглавила группу машинного перевода в ИТМ и ВТ АН СССР. Первый опыт перевода с Английского языка на русский с помощью машины БЭСМ был получен уже к концу 1955 г. Программы для БЭСМ составляли Н. П. Трифонов и Л. Н. Королев, кандидатская диссертация которого была посвящена методам построения словарей для машинного перевода.

Другое направление работ возникло в Отделении прикладной математики Математического института АН СССР (ныне ИПМ им. М. В. Келдыша РАН) по инициативе А. А. Ляпунова. К работам по машинному переводу математических текстов с французского языка на русский он привлек О. С. Кулагину, аспирантку МИАН, своих учениц Т. Д. Вентцель и Н. Н. Рикко. С конца 1955 г. в этих работах принимала участие Т. Н. Молошная, которая затем приступила к самостоятельной работе над алгоритмом англо-русского перевода. А. А. Ляпунов и О. С. Кулагина свои представления об использовании вычислительных машин для перевода с одного языка на другой опубликовали в журнале

"Природа",1955, № 8. Первые программы машинного перевода, разработанные этим коллективом, были реализованы на машине "Стрела".

Первое поколение систем машинного перевода базировалось на алгоритмах последовательного перевода "слово за словом", "фраза за фразой". Возможности таких систем определялись доступными размерами словарей, прямо зависящими от объема памяти компьютера. Перевод текста осуществлялся отдельными предложениями, смысловые связи между ними никак не учитывались. Такие системы называют системами прямого перевода. На смену им со временем пришли системы последующих поколений, в которых перевод от языка к языку осуществлялся на уровне синтаксических структур. В алгоритмах перевода использовался набор операций, позволяющий путем анализа переводимого предложения построить его синтаксическую структуру по правилам грамматики языка входного предложения (так же, как учат детей языку в средней школе), а затем преобразовать ее в синтаксическую структуру выходного предложения и синтезировать выходное предложение, подставляя нужные слова из словаря. Такие системы называются Т-системами (Т- от Английского слова "transfer - преобразование").

Наиболее совершенным считается подход к построению систем машинного перевода на основе получения некоторого, независимого от языков, смыслового представления входного предложения путем его семантического анализа. Затем производится синтез выходного предложения по полученному смысловому представлению. Такие системы называют И-системами (И - от слова "интерлингва"). Считается, что следующие поколения систем машинного перевода будут относиться к классу И-систем.

Как большой ученый, которому свойственно видеть всю проблему в целом, А. А. Ляпунов с самого начала работ по машинному переводу говорил о переводе путем извлечения смысла переводимого текста и его представления на другом языке. Однако такая постановка проблемы перевода оказалась в то время преждевременной. Более того, она не решена в общем виде мировой информатикой и в настоящее время, несмотря на усилия, предпринимавшиеся Международной федерацией IFIP - мировым сообществом ученых в области обработки информации. Однако многие частные результаты, связанные с семантическим анализом текстов, были получены и опубликованы в трудах IFIP.

Первый опыт создания программ машинного перевода показал, что необходимо решать эти задачи постепенно и по частям.

Слишком много трудностей и неясностей было в том, как нужно формализовать и строить алгоритмы для работы с текстами, какие сло-

вари надо вводить в машину, какие лингвистические закономерности следует использовать при машинном переводе и каковы вообще эти закономерности.

Выяснилось, что традиционная лингвистика не располагает ни фактическим материалом, ни идеями и представлениями, нужными для построения систем машинного перевода, которые использовали бы смысл переводимого текста.

Традиционная лингвистика не могла дать исходные представления не только в части семантики, но и в части синтаксиса. Ни для одного языка в то время не существовало перечней синтаксических конструкций, не были изучены условия их сочетаемости и взаимозаменяемости, не были разработаны правила построения крупных единиц синтаксической структуры из более мелких. В сущности ни на один вопрос, поставленный в связи с построением систем машинного перевода, традиционная лингвистика в 50-х годах не могла дать ответа.

Потребность в создании теоретических основ машинного перевода привела к формированию нового направления в лингвистике, называемого структурной, прикладной, математической лингвистикой. Формирование этого направления в СССР относится ко второй половине 50-х годов. Ведущую роль в нем сыграли математики А. А. Ляпунов, В. А. Успенский, (ученик А. Н. Колмогорова), О. С. Кулагина, лингвисты В. Ю. Розенцвейг, П. С. Кузнецов, А. А. Реформатский, И. А. Мельчук, В. В. Иванов.

6 мая 1960 г. было принято Постановление Президиума АН СССР "О развитии структурных и математических методов исследования языка", во исполнение которого были созданы подразделения по структурной лингвистике в Институте языкознания, Институте русского языка АН СССР. В Постановлении Президиума АН СССР отмечалось, что "недостаточное развитие теоретических исследований в области структурных и математических методов в лингвистических учреждениях тормозит практически важные работы по теории и практике машинного перевода, построению информационных языков и информационных машин, логической семантике и другим приложениям языкознания, разрабатываемым в настоящее время в ряде технических и математических научно-исследовательских институтов". С 1960 г. началась подготовка кадров в области автоматической переработки текстов на филологическом факультете МГУ, в Ленинградском и Новосибирском университетах, МГПИИЯ. Под математической лингвистикой понималось изучение языка как абстрактной знаковой системы с целью построения теоретической основы машинного перевода и создания конкретных алгоритмов

перевода. В таком понимании математическая лингвистика составляла часть семиотики - общей теории знаковых систем.

Задача аксиоматизации лингвистики была выдвинута одним из виднейших лингвистов московской школы П. С. Кузнецовым как задача формализации грамматики, восходящая к идеям выдающегося русского языковеда Ф. Ф. Фортунатова (1848-1914).

Исследованию формальной теории грамматик, была посвящена диссертация О. С. Кулагиной, выполненная под руководством А. А. Ляпунова.

Заметим, что в те же годы формальная теория грамматик развивалась в США в трудах Н. Хомского, ставших классическими для области искусственных языков, в частности языков программирования.

Двадцатилетие (1956-1976) один из основателей направления математик В. А. Успенский в своих воспоминаниях назвал "серебряным веком" структурной, прикладной и математической лингвистики в СССР (видимо, по аналогии с "серебряным веком" русской поэзии).

В 70-х годах разработку основ технологии машинного перевода продолжила группа специалистов в ВИНИТИ под руководством профессора Г. Г. Белоногова. В результате в 1993 г. была создана промышленная версия системы RETRANS фразеологического машинного перевода с русского языка на Английский и обратно, которая применялась в министерствах обороны, путей сообщения, науки и технологий, а также во ВНТИЦ.

Практическое применение принципов смыслового анализа текстов потребовалось при создании систем машинного перевода с иероглифических языков (китайского, японского и др.). Вопросы создания таких систем были разработаны в диссертации В. М. Зелко в 80-х годах.

Первые коммерческие продукты машинного перевода, нашедшие практическое использование, появились в середине 80-х годов. Они были реализованы на персональных компьютерах и являлись системами прямого перевода, возможности которых базировались на огромных (по сравнению с первыми системами) словарях, а не на умении анализировать и синтезировать тексты.

Современные коммерческие продукты машинного перевода предлагают отечественные фирмы: "Виста Текнолоджиз" и "Адвентис", образованные в 1991 г. коллективом разработчиков, выделившихся из ВИНИТИ; ПРОМТ, образованная в 1991 г.; "Медиа Лингва".

Однопользовательская "коробочная" версия продукта Retrans Vista фирмы "Виста текнолоджиз" предназначена для автоматизированного перевода текстов с русского языка на Английский и обратно. В ней ис-

пользованы оригинальные алгоритмы сжатия словарных баз и поиска переводных эквивалентов, позволяющих транслировать "на лету" не только фрагменты текста, импортируемые из текстового редактора MS Word, но и Web-страницы.

В словарях Retrans Vista хранятся миллионы понятий, к которым относятся не только традиционные устойчивые фразеологические обороты, но, прежде всего, словосочетания, используемые в повседневной речи. Кроме того, есть программа концептуального анализа, автоматически выделяющая из текста новые словосочетания и включающая их в словарь. Основные словари системы Retrans Vista содержат термины и фразеологические единицы по естественным и техническим наукам, экономике, бизнесу и политике. Объем политематического машинного словаря - около 3,4 млн. слов (1,8 млн. в русско-Английской части, 1,6 млн. - в англо-русской), причем 20% из них являются словами, а 80% - устойчивыми словосочетаниями со средней "длиной" в 2,2 слова.

Продукт Retrans Vista реализован на ПК с процессором, имеющим частоту от 166 МГц и ОЗУ от 32 Мб и выше и жестким диском от 170 Мб. Продукт работает под управлением ОС Windows 98/NT/2000.

Фирма ПРОМТ разработала и поставляет Интернет-переводчик PROMT Internet Translation Server, обеспечивающий перевод "на лету" Web-страниц, запросов к поисковым системам или к базам данных, представленным в Интернете.

Для корпоративных сетей многонациональных корпораций фирма ПРОМТ предлагает аналогичный продукт PROMT Intranet Server.

Модуль перевода PROMT Internet встраивается в браузер Microsoft Internet Explorer, образуя средство для синхронного перевода Web-страниц Web View. При этом можно устанавливать для перевода различные языковые пары: Английский - русский; Английский - немецкий; Английский - испанский; французский - Английский; французский - немецкий.

PROMT Internet Translator Server установлен на поисковой системе Voila, принадлежащей оператору France Telecom.

Для систем офисной автоматизации предлагается коммерческий пакет PROMT Lingvo OFFICE - результат сотрудничества двух лидеров российского рынка лингвистического программного обеспечения - ПРОМТ и ABBYY.

Компания "Медиа Лингва" выпустила электронные словари серии "МультиЛекс 3.5. Новый большой англо-русский словарь" и "МультиЛекс3.5. Английский. Экономика и право". Такие словари, работающие

под управлением операционных систем Windows CE или PalnOS, могут быть размещены на карманных компьютерах.

Классификация систем МП

Системы машинного перевода могут быть классифицированы на нескольких основаниях. Один из них тип лингвистических принятый стратегией в системе. С этой точки зрения выделяют четыре периода, каждый из которых характеризуется преимущественным развитием систем того или иного типа.

Начальный период “бурного развития” от конца 40 к середине 60-ых лет характеризуется преимущественным развитием прямых систем МП, реализующих лобовое решение проблемы перевода и обеспечивая результаты, близко к пословному переводу, так называемые системы первого поколения.

Второй период от середины 60-х к середине 70-х годов отмечен интенсивным развитием синтаксических теорий и разработкой на их основе СМП второго поколения.

Третий период от середины 70-х к середину 80-х годов можно назвать периодом экстенсивного развития СМП: они получили индустриальную жизнь. Техника морфологического и синтаксического анализа хорошо освоена, на повестке дня семантика. Но ожидаемого продукция к СМП третьего поколения, осуществляющего перевод через семантические структуры, не произошло. В качестве компенсации получают широкое развитие интерактивные СМП, комбинирующие работу человека и ЭВМ. Другим внешним решением семантических трудностей является ориентация на перевод ограниченных классов текстов, в частности, настроенных на узкой подчиненной области.

Четвертый период, так как вторая половина 80-ых лет, характеризуется резким увеличением интереса к МП как в практическом, так и в теоретическом плане: МП - сложная область, на которой выполнены новые информационные технологии. Появляется все больше многоязычных систем. Большие надежды возлагаются на мощные лексические и терминологические базы данных, базы знаний. Сближается проблематика МП и искусственный интеллект широкого профиля: в МП привлекаются семантические теории, созданные для экспертных и других систем искусственного интеллекта. Таким образом, СМП первого и второго периодов были противопоставлены прежде всего по типу лингвистической стратегии. В системах первого и полуторного поколений операция перевода требует минимума преобразований: исходный текст постепенно превращается в текст на выходном, языке путем замены

всех его элементов, найденных в словаре, на переводные эквиваленты, никакая языковая модель не требуется, кроме переводных в основном лексических и позиционных соответствий. Учитывается лишь локальный контекст, он же позволяет собирать некоторые сложные единицы - обороты, поэтому такой перевод называют пословным, или пословно-пооборотным. Для систем данного типа характерны бинарность, отсутствие промежуточных структур, анализ ведется сразу в категориях выходного языка, одно вариантность. Их сильная сторона - простота устройства и вытекающая отсюда большая скорость работы.

В системах второго поколения переводные соответствия устанавливаются не "прямым" путем, а но только после для каждого предложения выявлена в результате анализа его синтаксическая или синтактико-семантическая структура или несколько альтернативных вариантов такой структуры. Анализ и синтез являются независимыми, анализ, как правило, многовариантный, ведется в категориях входного языка, синтез - в категориях выходного, так что связь того и другого этапов обеспечивается введением особого этапа межъязыковых операций.

На этом, которое лингвистические основания классификации СМП прервались, как перевод через семантический посреднический языком, универсальный для разных пар естественных языков, не был обеспечен единой общепризнанной лингвистической теорией. Определение и развитие СМП третьего и четвертого поколений остается делом будущего. Системы пятого поколения выделены в японском проекте по другому основанию: они должны стать частью компьютеров пятого поколения. Это развитые многоязычные системы, базирующиеся на самой передовой информационной технологии. Их лингвистические возможности предполагается расширить освоением речевого ввода и вывода.

В литературе 80-ых лет этим более говорят о делении СМП не на поколениях, а на системы с прямым и непрямым переводом, а последних - на системы с трансфером и с языком-посредником. Иногда СМП классифицируют на синтаксически ориентированные и лексически-ориентированные или СМП, работающие под управлением словаря - они приближаются по технике анализа к системам класса искусственного интеллекта.

В 80-е годы в отдельный класс выделяют СМП, основанные на знаниях. В системах этого класса, представляющий подкласс систем искусственного интеллекта, в качестве отдельного компонента включаются экстралингвистические знания, хотя они могут иметь те же формы представления, что и собственно лингвистическая информация, т. е. записываться в словарях и грамматиках. Отчетливо к этому классу при-

надлежат те СМП, которые используют концептуальную сеть знания при анализе.

Логико-алгоритмические основания, современные больше наиболее развитые СМП обычно настолько гибки, который может включить любую лингвистическую теорию и допускают разные их комбинации. Однако они имеют, как правило, собственную модель процесса перевода. Следующие два основания классификации примыкают к лингвистическому. По количеству привлекаемых языковых пар СМП делятся на двуязычные (реализующие функцию перевода только для данной языковой пары) и многоязычные. Те и другие в зависимости от техники лингвистического анализа могут быть либо бинарными, если анализ входного языка ведется в категориях выходного, либо универсальными, если устройство анализа не зависит от выходного языка. Универсальная двуязычная СМП может легко стать многоязычной при комбинации с компонентами анализа и/или синтеза других универсальных систем. Система же, состоящая из совокупности бинарных СМП, может быть названа многоязычной, но не является универсальной.

По тематической ориентации отличают системы моно тематические, настроенные на одну ПО, таких большинство: TAUM-METEO, SPANAM, METAL, TITUS, и политематические. Иногда СМП имеют ограничения на структуру вводимых текстов, СМП с ограниченным ЕЯ, например, система TITUS или TITRAN, переводящая только заголовки.

Классификация СМП может рассмотреть также технические характеристики: масштаб, степень реализованности, доля участия человека в процессе МП.

С практической точки зрения, имея в виду качество результирующего текста и его соответствие исходному, программы машинного перевода подразделяют на три категории:

- полностью автоматический перевод;
- автоматизированный машинный перевод при участии человека;
- перевод, осуществляемый человеком с использованием компьютера.

Программы машинного перевода первой из названных категорий являются делом далекого будущего, поскольку в общем виде не решены проблемы автоматического понимания, перевода и синтеза текстов.

Программы второй категории разработчики называют МТ-программы. Реально автоматизированный машинный перевод возможен только в условиях искусственно ограниченного, как по словарному запасу, так и по грамматике, языка.

В качестве реального успешного проекта МТ-программы всегда называют немецкую систему Meteo, выполняющую перевод метеопрогнозов с французского языка на Английский и обратно.

К МТ-программам относятся и продукты машинного перевода компании ПРОМТ, упомянутые выше, в том числе программы для просмотра содержимого Web-страниц в сети Интернет с целью поиска нужного документа.

Программы третьей категории разработчики называют ТМ-программы (от translation memory - память перевода). Эту категорию программ применена профессиональными переводчиками, осознавшие выигрыш от автоматизации их работы с помощью компьютеров. Основу ТМ-программ составляют специализированные словари, соответствующие тематике переводимого текста. При переводе используются конструкции и значения слов и устойчивых словосочетаний, выбранные профессиональным переводчиком и занесенные в словари системы, а полученный текст подвергается интенсивному редактированию. Словари и уже переведенные фрагменты текстов, запоминаемые в ТМ-системе, могут быть повторно использованы в больших коллективных проектах, ими можно обмениваться. Поэтому ТМ-системы представляют собой важное средство автоматизации труда профессиональных переводчиков.

Часто ТМ-программы используют в сочетании с МТ-программами. Наиболее популярным в мире ТМ-инструментарием является Translation's Workbench фирмы Trados, для краткости часто также называемый Trados.

За 17 лет своего существования фирма Trados продала 45 тысяч лицензий на свою систему. Все они приобретены профессиональными переводчиками. В конце 2001 г. Российская фирма ПРОМТ, известная своими продуктами машинного перевода категории МТ, объявила о получении статуса эксклюзивного дистрибьютера системы Trados в России и других странах СНГ. Для совместного использования своих МТ-программ и продуктов Trados фирма ПРОМТ предлагает специальные средства их сопряжения.

История машинного перевода насчитывает немногим более 50 лет. В течение этого времени некоторые поколения систем машинного перевода - от первых программ, использовавших ограниченные ресурсы универсальных компьютеров первого поколения до современных коммерческих продуктов, использующих мощные ресурсы серверов и персональных компьютеров, включая ПК, в которых можно размещать карманные словари, а также компьютерные сети.

По мере снятия технических ограничений, налагаемых возможностями компьютеров по производительности и памяти, становилось ясно,

что проблема перевода текста с одного естественного языка на другой принципиально не сводится только к перекодировке слов. Для преодоления основных трудностей проблемы машинного перевода должны быть решены задачи автоматизированного представления контекста, смыслового содержания переводимого текста, знаний о понятиях предметной области, к которой относится переводимый текст.

Вместе с тем современные достижения в области вычислительной техники, информационных технологий и технологий телекоммуникаций позволяют выдвигать на перспективу практические задачи поиска и выбора требуемой информации, представленной на разных языках, из разнородных источников, находящихся в корпоративных и глобальных информационно-телекоммуникационных сетях.

В качестве примера такой перспективной задачи можно привести системы запросов к информационным ресурсам сетей, например к базам данных, с возможностью формирования ответов по телефону в виде устной речи. Для этого требуется сочетание систем машинного перевода с системами распознавания и синтеза речи.

Постановка задачи

От предыдущей части мы уже видели история развития различные поколения машинного перевода в течение различного времени, и можем быть постановить следующие задачи:

Изучение основных элементарных структур машинного перевода и найти одну самую хорошую структуру для Русско-Вьетнамского МП, которая может обеспечить систему не только хорошо переводить и еще может непрерывно развиться

Качество МП очень зависит от качества базы данных, которая представляет собой набором словаря, поэтому мы будем строить несколько основных словари от них. Изучение методов работы между словарями и процессами перевода, и от данного результата мы знаем какие словари являются необходимыми, будем строить, их структура и составлять в среднем объем.

В настоящем время бывает несколько фирм уже успеха делать МП и получит прибыл от них поэтому мы тоже изучим бизнес - план разработки набора словаря русско-вьетнамского машинного перевода, рассмотрены вопросы организации и планирования процесса разработки, а также определение стоимости и цены программного продукта, эффективности его разработки и внедрения.

Основные этапы машинного перевода

Общая структура машинного перевода

Компоненты машинного перевода, составляющего языковую модель, - лингвистические процессоры, которые друг за другом обрабатывают входной текст. Вход одного процессора является выходом другого. Выделяются следующие компоненты:

Графематический анализ разделяет входной текст на предложения и предложения на слова.

Морфологический анализ осуществляется морфоанализ и лемматизация русских словоформ, формирование граммем слов.

Фрагментационный анализ текста. Разделение сложного предложения, на несколько простых. Путём распознавания, где в предложении стоят союзы и разделительные знаки препинания

Синтаксический анализ- К полученным простым предложениям, данный модуль, используя собственный компилятор, подбирает подходящие грамматики из БД, на основе которых строится синтаксическое дерево.

Семантический анализ. Построение семантического графа текста.

Для каждого уровня разрабатывался свой язык представления. Язык представления, поскольку это необходимо, состоит из констант и правила их комбинации. На графематическом уровне константами были графематические дескрипторы, пример: ЛЕ – лексема, ЦК – цифровой комплекс и т.д. На морфологическом уровне – граммемы, пример: рд – родительный падеж, мн -множественное число. На синтаксическом – названия отношений и групп, пример: ПОДЛ – отношение между подлежащим и сказуемым, ПГ - предложная группа. На семантическом – семантические категории и отношения.

С каждого уровня представления можно сделать переход к тому же самому представлению на другом естественном языке (трансфер), которая позволяет осуществлять перевод, даже если "глубокий" (семантический) анализатор не смог обработать текст. Основой для построения уровней служили результаты работы предыдущих этапов, но, что важно, последующие анализаторы также могли улучшить представление предыдущих. Например, для какого-то предложения синтаксический анализатор не смог построить полного дерева зависимостей, тогда, возможно, семантический анализатор сможет спроектировать им построенный семантический граф на синтаксис. Таким образом, текст обрабатывается по следующей технологии:

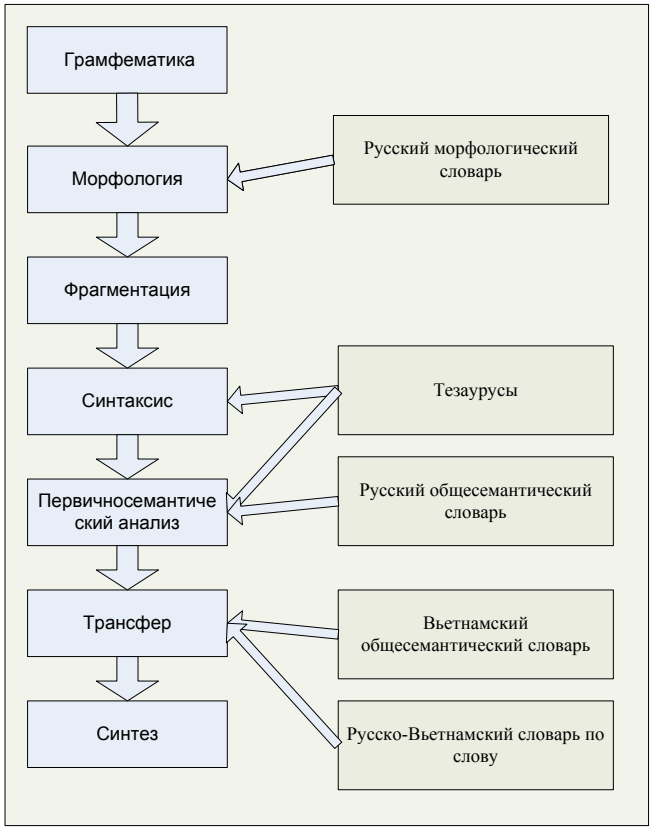


Рис 1. Основные элементы машинного перевода.

Мы знаем, что, объект машинного перевода - текст естественного языка, в который количество слова большое, и много синтаксических правил и понятий. Поэтому машинный перевод - сложная система, состоит из многих процессов и большого набора словаря, и чем больше качества и количества слова каждого словари, чем лучшее система переводит текст.

Графематический анализ

(http://www.aot.ru/docs/graphan.html)

Графематический анализ (ГрафАн) - это программа начального анализа естественного текста, представленного в виде цепочки ASCII символов, вырабатывающая информацию, необходимую для дальнейшей обработки Морфологическим и Синтаксическим процессорами. В задачу графематического анализа входят:

1. Разделение входного текста на слова, разделители и т.д.
2. Сборка слов, написанных в разрядку;
3. Выделение устойчивых оборотов, не имеющих словоизменительных вариантов;
4. Выделение ФИО (фамилия, имя, отчество), когда имя и отчество написаны инициалами;
5. Выделение электронных адресов и имен файлов;
6. Выделение предложений из входного текста;
7. Выделение абзацев, заголовков, примечаний.

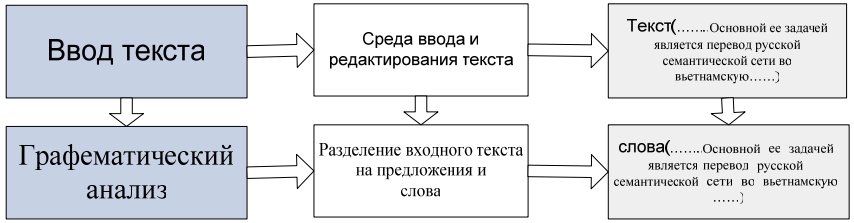


Рис 2. Графематический анализ

Морфологический анализ

Осуществляется морфоанализ и лемматизация русских словоформ, формирование граммем слов.

Лемма – это нормальная форма слова. Например, для существительных – это единственное число (если оно есть у существительного), именительный падеж. То есть лемматизация - приведение текстовых форм слова к словарным; а морфоанализ приписывание словоформам морфологической информации.

Данную задачу не представляет труда выполнить для русского языка благодаря Его развитой морфологии практически со стопроцентной точностью.

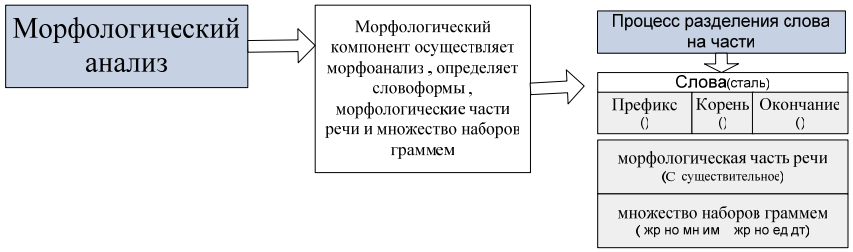


Рис 3. Морфологический анализ

Фрагментационный анализ

Задача фрагментационного анализа состоит в выделении в предложении синтаксических единств (фрагментов). Разделение сложного предложения, на несколько простых. Путём распознавания, где в предложении стоят союзы и разделительные знаки препинания.

Первая важная особенность фрагментов заключается в том, что их границы не пересекают синтаксические связи, соединяющие отдельные слова или словосочетания. Таким образом, при успешной работе фрагментационного анализа перед синтаксическим исключается возможность построения большого числа неправильных синтаксических связей, которые допускаются морфологией, синтаксисоми, возможно, семантикой.

Второе важное свойство – членение предложения на фрагменты в большинстве случаев соответствует его делению на крупные семантические узлы. Так, результат фрагментационного анализа несет и некоторую семантическую информацию о предложении

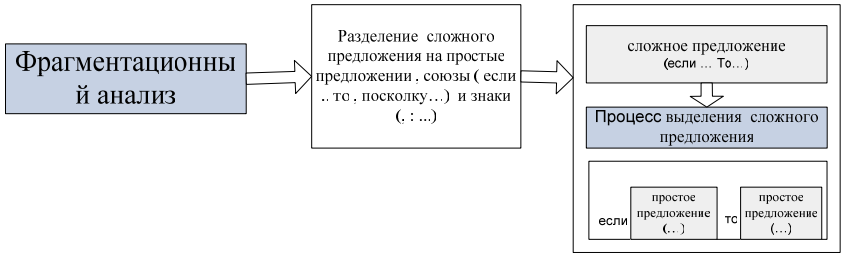


Рис 4. Фрагментационный анализ

Синтаксический анализ

Цель синтаксического анализа -автоматическое построение функционального дерева фразы, т.е. нахождение взаимозависимостей между разноуровневыми элементами предложения. На вход синтаксису подаются результаты морфологического анализа (каждой словоформе сопоставлено максимально возможное для данной словоформы множество морфологических интерпретаций) и фрагментационного анализа

К полученным простым предложениям, данный модуль, используя собственный компилятор, подбирает подходящие грамматики из БД, на основе которых строится синтаксическое дерево.

Например:

Простое предложение: *Я иду в школу.*

Грамматика: *сущ. + глаг.+предл.+сущ.*

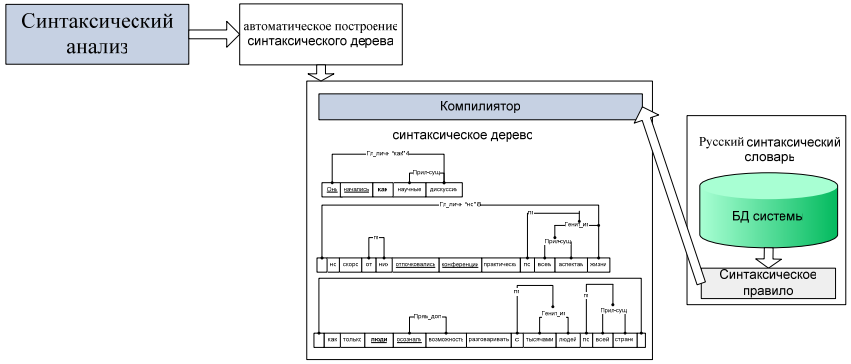


Рис 5. Синтаксический анализ

Семантический анализ

В отличие от синтаксического анализа семантический этап использует формальное представление смысла составляющих входной текст слов и конструкций. В сферу семантического анализа входит:

Построение семантической интерпретации слов и конструкций;

Установление «содержательных» семантических отношений между элементами текста, которые уже принципиально не ограничены размером одного слова (могут быть больше или меньше одного слова).

Можно сказать, что создание полных систем машинного перевода для русского языка, использующих семантический анализ, является чрезвычайно актуальной задачей. Поэтому, в следующей главе подробнее остановимся именно на семантическом анализе русскоязычных текстов на естественном языке.

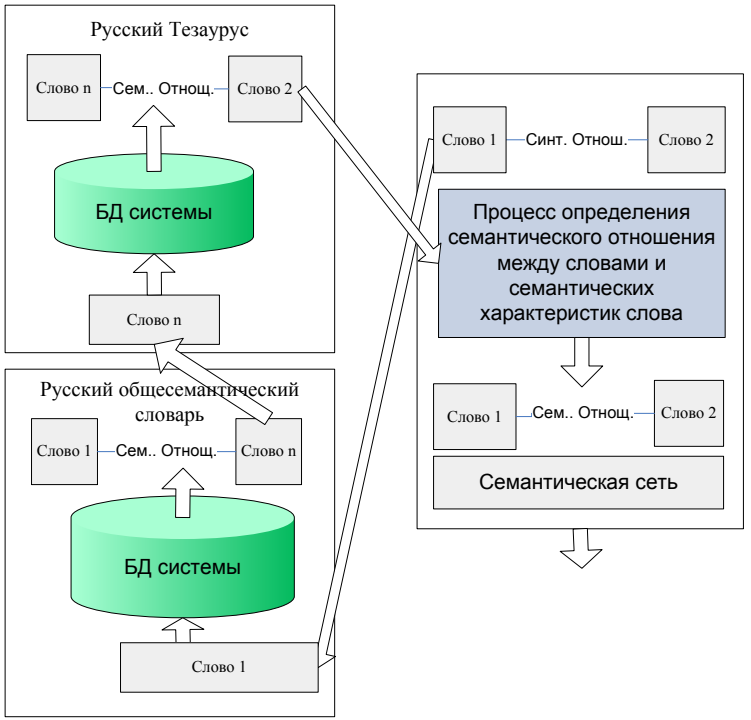


Рис 6. Семантический анализ

Трансфер и синтез

На этапе *трансфера* решаются следующие задачи:

1. Для узлов русского семантического графа ищутся Вьетнамские эквиваленты по Вьетнамскому словарю, из которых строятся Вьетнамские семантические узлы;
2. Вьетнамские семантические отношения строятся по русским отношениям с необходимыми перестройками;
3. Актантам Вьетнамских узлов приписываются грамматические характеристики в соответствие с их словарными статьями.

На этапе *синтеза*, работающим непосредственно после этапа трансфера, осуществляется следующее:

1. Порождаются Вьетнамские словоформы по заданным на этапе трансфера граммемам;
2. Определяется порядок слов;

3. Осуществляется перевод терминов, групп времени, слов, не вошедших в семантические словари;
4. Синтезируются артикли для именных групп.

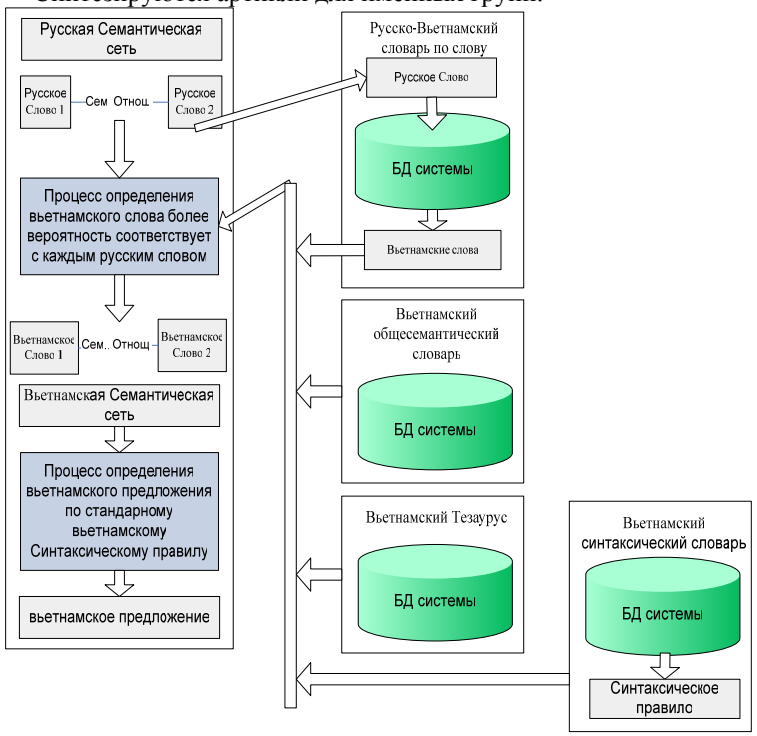


Рис 7. Трансфер и синтез

Литература

http://www.computer-museum.ru/histsoft/histmt.htm История машинного перевода Е. Н. Филинов

ю. и. шемакин начала компьютерной лингвистики москва – 1992

http://www.langust.ru/rus_gram/rus_gr03.shtml#rg03_15

http://www.aot.ru/docs/rusmorph.html

АОТ Технологии Русский общесемантический словарь.htm

М.А.Павлова

ЭЛЕКТРОННЫЕ БИБЛИОТЕКИ В СОВРЕМЕННОМ ИНФОРМАЦИОННОМ ОБЩЕСТВЕ

Электронные библиотеки в современном информационном обществе по мере развития программно-технических и телекоммуникационных средств приобретают все большее значение во всех сферах деятельности общества. Они, кроме того, оказывают существенное влияние и на изменение технологии работы традиционных библиотек, использующих автоматизированные библиотечно-информационные системы. Электронные технологии дают возможность для практически безграничного расширения увеличения пользователей любой социальной принадлежности. Библиотеки приобретают возможности и постепенно входят в процесс реализации информационного сопровождения таких пользователей.

На сегодняшний день существует множество понятий и значений, описывающих термин электронной библиотеки в библиотечно-информационном деле. Электронные библиотеки, являющиеся хранилищами знаний, можно рассматривать как сложные информационные системы, при создании и использовании которых требуется решение многих научных, технологических, методологических, экономических, правовых и других вопросов

Под электронной библиотекой можно понимать также собрание переведенных в цифровой формат тексты уже существующих и выпущенных на свет книг, а также примечаний, которые можно встретить в книгах помимо самих текстов. В электронную библиотеку также могут входить журналы, в том числе и иллюстрированные, и отдельные статьи.

Одна из проблем с понятием «электронная библиотека» связана с тем, что употребление термина «библиотека» уже само по себе дезориентирует: именно из традиционного представления о том, что такое библиотека, вытекает отношение к электронной библиотеке как к расширению функциональных возможностей обычной библиотечной автоматизированной системы, к которой лишь добавляются функции работы с оцифрованными документами. В силу этого все предпочитают, чтобы электронными библиотеками занималось библиотечное сообщество.

Отсюда вытекает и бытующее неправильное понимание данного феномена: электронная библиотека – это та же библиотека, только в ней все электронное – и документ, и услуги, и «читатель».

С возрастанием запросов на информацию в период современного развития информационного общества, с привлечением к ней все более широкого круга людей возрастает потребность в совершенствовании работы библиотек. Они превращаются в общественные информационные центры широкого назначения, в обеспечении их квалифицированными информационными работниками, посредниками между производителями и пользователями информацией. При этом библиотеки будут успешно играть роль важнейшего элемента управления информационными ресурсами общества, если:

1) электронные библиотеки смогут обеспечивать возможность качественного управления электронными потоками информации;

2) электронные библиотеки смогут эффективно работать в глобальном информационном пространстве, отбирая необходимую нашему обществу информацию;

3) электронные библиотеки смогут выступать в качестве центров надежного хранения новой электронной и переведенной с других носителей в электронную форму информации;

4) переведенная в электронную форму информация фондов библиотек будет вводиться в активный оборот, создавая единые массивы с новой, утверждая наработанные предыдущими поколениями традиции создания информации;

5) за счет планомерного комплектования фондов будут восполнены массивы информации, способствующие укреплению междисциплинарных связей в науке, укреплению всей системы информационных баз общества;

6) библиотеки, активизируя работу по удаленному обслуживанию пользователей, будут отрабатывать технологии, направленные на повышение эффективности использования информации, в том числе и путем адаптации к потребностям заказчика, с учетом специфики использования данной информации;

7) библиотеки начнут процесс вхождения на информационные рынки в качестве полноправных субъектов, что будет иметь для них не только материальные стимулы;

8) библиотеки в качестве общественных информационных центров станут также центрами передового опыта, научной мысли, просвещения и образования для всех категорий граждан, которые используют совре-

менные информационные технологии в качестве важного элемента своего образа жизни.

Опишем основные отличия традиционной библиотеки от электронной:

1) самое принципиальное отличие – это конечно разные виды носителей, твердые и машиночитаемые;

2) технологические процессы формирования информационной продукции, а также составом средств, используемых для ее хранения и предоставления пользователям;

3) характер целей и решаемых задач обслуживания пользователей;

4) организационные формы создания и функционирования – традиционные библиотеки являются юридическими субъектами или структурными службами организаций (фирм, предприятий), в состав которых они входят. Они имеют профессиональный штат сотрудников и предназначены обслуживать литературой определенные контингенты читателей. Электронные библиотеки создаются или могут создаваться в рамках организаций любого вида, имеющих, как правило, собственные информационные ресурсы, которые они заинтересованы и способны предоставить в доступ пользователям через Интернет или специализированные телекоммуникационные сети. При этом основной состав исполнителей, поддерживающих их работу, в большинстве своем профессионально не ориентирован на выполнение библиотечных и информационных задач, которые являются для них дополнительными или временными;

5) разные права собственности на информационную продукцию, а также на ее копирование и распространение – ЭБ пользуются такими правами в силу специфики документов и данных, которыми они владеют и предоставляют в доступ, традиционные библиотеки правами собственности на литературу, которой они располагают, как правило, не обладают;

6) различный способ обращение пользователей к ресурсам и услугам этих библиотек. Чтобы получить нужную литературу читатель обычной библиотеки, как правило, приходит в нее. Он является постоянным пользователем ее услуг и для обращения к ресурсам библиотеки собственных программно-технических средств ему не требуется. Поиск нужной ему литературы он производит во вполне определенном для него электронном каталоге и, при необходимости, может воспользоваться помощью сотрудника библиотеки. Пользователь электронной библиотеки обращается ко всему пространству Интернета для получения необходимых ему документов и данных. В этих условиях, как правило, он не

вправе предъявлять какие-либо претензии персоналу электронной библиотеки в недостаточном внимании или качестве и обслуживания.

Среди достоинств электронной библиотеки обычно указываются ее доступность и удобство использования. Действительно, электронная библиотека, в отличие от обычной, не имеет графика работы. Тексты библиотеки доступны в любое время дня и ночи, количество копий не ограниченно. В электронной библиотеке все книги равнодоступны для пользователя, что не всегда бывает в обычной библиотеке. Кроме того, тексты не нужно возвращать в библиотеку к определенному сроку.

Принцип, лежащий в основе электронных библиотек, можно возвести к глубокой древности. Некоторые называют этот принцип александрийским – по названию Александрийской библиотеки. Суть его состоит в том, что зафиксированная информация хранится в виде одного материального объекта (документа) в одном месте, все желающие имеют к ней доступ и могут по мере необходимости ее копировать для своих потребностей.

Среди первых работ, в которых было предсказано появление электронных библиотек и описаны их общие принципы, обычно называют статью В. Буша «Как мы можем думать» (1945) и книгу Дж. С. Р. Ликлидера «Библиотеки в будущем» (1965). Предложенная А. Бушем концепция информационной системы, названной им «Memex», базировалась на использовании фотографий для хранения информации и в определенном смысле предвосхитила дальнейшее изобретение и внедрение микрофильмов.

Зарождение электронных библиотек относится к концу 80-х годов, когда стали создаваться первые электронные библиотеки научных журналов (проекты «Mercury», CORE, «Tulip», 1987-1993 гг.; JSTORE, с 1995 г.; «High Wire Press», с 1995 г., и др.). Эти проекты преследовали как научные, так и экономические цели (создание архива важнейших журналов и обеспечении широкого доступа к нему, сокращение расходов библиотек за счет устранения дублирования коллекций журналов).

Кроме того, крупные библиотеки и музеи приступили к оцифровке хранящихся у них материалов, прежде всего редких, старинных и находящихся под угрозой физического разрушения, чтобы сохранить их для будущего и сделать общедоступными. Такие проекты, получившие название конверсионных. Примерами проектов такого рода могут служить программы «American Memory» (с 1989 г. по настоящее время) и «National Digital Library» (с 1990 г., в 1998 г. преобразована в единую межведомственную программу – «Digital Libraries Initiative-Phase 2»), целью которых является перевод в электронную форму материалов,

значимых для истории и культуры США. С середины 90-х годов подобные проекты стали осуществляться и в России (оцифровка коллекций Эрмитажа, редких рукописей в РГБ, ВГБИЛ и т. д.).

На начальных этапах становления Интернета значительный вклад в построение электронных библиотек внесли любители-энтузиасты, создавшие большое число ресурсов, некоторые из которых получили весьма широкую известность. Наиболее известными проектами такого рода являются «Проект Гутенберг», инициированный в 1990 г. американским программистом М. Хартом, и российская «Библиотека Мошкова», запущенная в ноябре 1994 г. и к августу 2001 г. включавшая в себя более 17 тыс. текстовых файлов общим объемом более 2,5 Гб и увеличивающаяся по сей день.

При всей несомненной общественной значимости любительские проекты обладают рядом существенных недостатков по сравнению с профессиональными электронными библиотеками:

– стихийность формирования фондов, неясность принципов отбора;

– случайность и неполнота собраний;

– недостаточная текстологическая база: произвольные источники публикации, опечатки, отсутствие необходимой библиографической информации; отсутствие справочно-комментаторского аппарата;

– технологическая примитивность: «слабая» разметка документов (текст ASCII или простой HTML), минимальное количество сервисов для читателей.

Все это приводит к тому, что любительские электронные библиотеки выполняют общекультурную функцию, выступая как удобный источник текстов в электронной форме, однако использовать эти библиотеки в научных и образовательных целях полноценно, как правило, невозможно.

Большая часть электронных библиотек в США и Западной Европе возникла и развивается в рамках академических и исследовательских организаций, к которым относятся, прежде всего, университеты. Именно в университетах и в университетских библиотеках и издательствах (при финансовой поддержке со стороны государства, благотворительных фондов и корпораций) осуществляются передовые исследования в области электронных библиотеках, разрабатываются стандарты и создаются разного рода цифровые коллекции.

Многие университетские ЭБ, однако, предоставляют лишь ограниченный доступ – либо только своим сотрудникам и студентам, либо некоторому числу других университетов и культурных учреждений, получая в обмен доступ к их цифровым коллекциям. Примером может слу-

жить Калифорнийская цифровая библиотека, предоставляющая доступ к значительному числу электронных коллекций и баз данных в девяти кампусах Университета Калифорнии.

Что касается университетов в России вплоть до последнего времени, академические организации в этом отношении были довольно пассивны. Дело, как правило, ограничивалось созданием тематических коллекций текстов (по математике, психологии, общественным наукам и т. п.), причем по технологическому уровню публикаций эти коллекции были сопоставимы с любительскими, значительно уступая им в охвате и темпах роста. В ряде случаев функции университетов принимают на себя независимые публикаторы. В качестве примеров можно назвать проект «Общий текст», созданный, «чтобы сообща пополнять Сеть отсутствующими в ней текстами», и «Русскую виртуальную библиотеку» – первую в русской Сети библиотеку академического типа, публикующую русскую классику по авторитетным изданиям с приложением справочного аппарата и комментариев.

Необходимо отметить, что на Западе созданием и поддержкой электронных библиотек и других электронных продуктов занимается целый ряд коммерческих компаний, продающих как сами эти продукты на жестких носителях, так и лицензии на доступ к цифровой информации online. Так, компания «ProQuest», базирующаяся в Анн-Арборе (штат Мичиган, США) и являющаяся одним из крупнейших провайдеров информационных услуг, лицензирует доступ к полным текстам тысяч периодических изданий, диссертаций, книг и других публикаций включая обширные электронные ресурсы компании «Chadwyck-Healey», которая в 1999 г. стала подразделением «ProQuest».

Впечатляющий пример национальной программы создания электронной библиотеки дает Япония, где в 1989 г. было начато проектирование электронной «библиотеки XXI в.». Для проведения исследований, экспериментов и подготовки проектной документации было создано специальное Агентство по внедрению новых информационных технологий (IPA). В его состав вошли специалисты из Национальной парламентской библиотеки, Национального центра НТИ, других информационных центров, а также из крупных фирм, действующих на рынке информационных технологий. В результате был подготовлен проект Электронной библиотеки Японии в провинции Кансай (500 км от Токио), преобразовано в электронную форму (главным образом в виде изображений) более 10 млн страниц различных печатных изданий (книги, журналы, газеты, карты и др.). В марте 1998 г. был подготовлен план реализации всего проекта со сроком завершения в 2003 г. Общая стои-

мость проекта оценивается в 500 млн. долл. К его реализации привлечено девять крупных фирм, в том числе NEC, «Mitsubisi», «Fujitsu».

В России же дела с электронными публикациями в библиотеках обстоят не блестяще. Ни одна из крупных библиотек до сих пор не породила ничего достойного внимания или могущего конкурировать с любительскими электронными библиотеками. Как правило, библиотечные проекты в этой области являют собой печальные образцы «освоения грантов»: создается какой-нибудь «каталог ресурсов», и на этом дело заканчивается. Примером такого рода проектов может служить «Виртуальная библиотека» при ГПНТБ России, созданная, как указано на сайте, «при поддержке Совета по программе «Информатизация России». В Виртуальной Библиотеке можно отыскать необходимые ссылки на ресурсы Internet. Представлено более 2000 адресов онлайновых журналов, газет, WWW сайтов и домашних страниц». Есть в этой библиотеке и собственные онлайновые публикации, правда, совсем немного.

Значительное количество ЭБ в России создается силами отдельных предприятий, организаций и учреждений.

Public.ru: «Публичная интернет-библиотека». Поддерживается ЗАО «Публичная библиотека». Представляет собой базу данных по российским СМИ. Основной фонд составляют публикации российских периодических изданий с 1990 г. по настоящее время. Предоставляется свободный доступ, но предлагаются и платные услуги: обслуживание в режиме «профессиональный поиск», подбор документов по заданным критериям, составление тематических баз данных, составление на заказ справок и аналитических материалов, подписка на тематические обзоры прессы.

«Национальная электронная библиотека» (НЭБ) – электронный архив русскоязычных открытых информационных источников. Возник как внутренний проект Национальной службы новостей в 1994 г. и по мере развития пополнялся за счет сбора текущих публикаций и покупки существующих информационных архивов. Основные темы: банковская деятельность, внутренняя и международная политика, культура, преступность, макроэкономика, прогнозируемые отставки и назначения во властных центральных и региональных структурах. Все услуги платные.

«Научная электронная библиотека». Владелец: ООО «Интра-Центр+». Спонсоры: Российский фонд фундаментальных исследований, Фонд Сороса, Министерство образования. Предоставляет доступ к зарубежным и российским научным журналам и базам данных крупным российским университетам, научно-техническим библиотекам России и их филиалам, части научных библиотек институтов РАН и региональ-

ных научных библиотек России. Доступ ограничен, требуется институциональная подписка.

«Интегрированная система информационных ресурсов» (ИСИР). Разработчик: Центр научных телекоммуникаций Российской академии наук (ЦНТК РАН). Система предназначена «для обеспечения доступа ученым, научным коллективам и организациям к информационным и вычислительным ресурсам РАН, организации оперативного обмена научной информацией и создания на основе современных информационных технологий условий для проведения совместных исследовательских работ», а также для создания единой системы описания публикаций, осуществляемых в рамках РАН.

Фундаментальная электронная библиотека «Русская литература и фольклор» (ФЭБ). Совместный проект НТЦ «Информрегистр» и Института мировой литературы им. А. М. Горького РАН. Представляет собой «сетевую многофункциональную информационную систему, аккумулирующую информацию различных видов (текстовую, звуковую, изобразительную и т. п.) в области русской литературы XI–XX вв. и русского фольклора, а также истории русской филологии и фольклористики. Цель проекта – представить важнейшие произведения русской словесности согласно принципам научных изданий: публикация по авторитетным источникам, наличие вариантов, обширный справочный и библиографический аппарат и т. д. Все материалы находятся в открытом доступе.

С середины 90-х годов в России осуществлялся ряд государственных целевых программ, в той или иной степени направленных на развитие электронных библиотек для нужд науки и образования: федеральная целевая программа «Электронная Россия на 2002—2010 годы», федеральная целевая программа «Развитие информатизации в России на период до 2010 г.», федеральная целевая программа «Развитие единой образовательной информационной среды на 2001—2005 гг.»

Интенсивные работы по созданию электронных библиотек, а также по выработке общей концептуальной и технологической базы в этой сфере ведутся в РГБ, ВГИБЛ, ЦНТК РАН, НТЦ «Информрегистр» и ряде других организаций.

Остро стоит вопрос в современном информационном обществе о переходе традиционных библиотек в электронную форму, то есть возможность существования этих двух форм одновременно. Но в рамках сегодняшней действительности эта задача решается малорезультативно. Рассмотрим этот вопрос. Первый способ – планшетный сканер среднего качества, с которого начинается перевод текста в оцифрованный вид,

стоит копейки – от 40$ до 70$. Программное обеспечение для распознавания текста в России не составит труда найти даром. Проблема стоит в низких объемах сканирования, а фонды содержат порядка сотен тысяч книг и если нужно обеспечить их сохранность, а в год сканируется сто, то и говорить об этом не стоит. Принимая решение о сканировании фонда документов, необходимо решить каким образом будет осуществляться этот процесс: какие части фонда и в какой последовательности будут подвергаться сканированию; в каком виде будет храниться результаты сканирования – в виде текста или в виде изображений; в каком формате тексты будут представляться в Internet – html, pdf, другом и т. п.

Возникает вопрос, в какой степени традиционные библиотеки должны сами заниматься переводом своих бумажных экземпляров в электронную форму? Этот вопрос решается прагматически – нужно найти спонсора, гранты на оцифровку, причем нужно иметь обоснование, что оцифрованная коллекция будет пользоваться спросом и по крайней мере покроет затраты на ее создание. Если речь идет о национальном информационном ресурсе, то стоит к этому вопросу привлечь государственные органы и добиться бюджетного финансирования, но на сегодняшний день, подобное решение находиться в пределах мечты.

Второй способ – получение электронных копий документов непосредственно от издательств (или от авторов). Это наиболее простой и экономичный способ, требующий при этом решения ряда технических и организационных вопросов. Например, в каком формате будет осуществляться передача документов, какой дальнейшей обработке будут подвергаться эти документы, что с ними будет происходить в дальнейшем. Определить, на каких условиях электронные документы будут передаваться библиотеке.

По мере развития электронных библиотек возникает проблема разработки стандартов для них, в том числе касающиеся структуры данных, сопровождающих электронные публикации. Хотя в последнем случае могут использоваться некоторые уже существующие прописи СИБИД, но они должны быть выполнены с учетом того, что они связаны с принципиально отличными от традиционных объектов нормирования, условий их распространения и использования. В создании «института стандартизации» для электронных библиотек нет необходимости, поскольку в соответствии с установленным в России порядком, разработкой стандартов любого рода занимается ВНИИКИ. Другой вопрос заключается в том, что инициативу выполнения таких разработок до сих пор не взяло на себя ни одно ведомство нашей страны.

В заключение заметим, что развитие электронных библиотек ввиду всеобщей компьютеризации является очень важным событием. Люди меньше ходят в традиционные бумажные библиотеки, особенно если учесть современные решения проблем безопасности устройств отображения, они замечают, что в электронной библиотеке легче производиться поиск требуемой информации, и имеется возможность получить быстрый и своевременный доступ к информации в любое время суток. Хотя нужно заметить, что навряд ли традиционная библиотека исчезнет в ближайшее время с лица Земли, поскольку в ее состав входят профессиональные специалисты библиотечного дела, и одна из их самых важных задач – каталогизация.

Литература

1. *Нохрин Ю.В.* Анализ терминоситемы «Электронная библиотека» // http://libconfs.narod.ru/2004/s1/s1_p18.htm;
2. *Горный Е.* Развитие электронных библиотек: мировой и российский опыт, проблемы, перспективы. // http://www.zhurnal.ru/staff/gorny/texts/dlib.html;
3. *Хохлов Ю.Е.* О месте электронных библиотек в информационном обществе. // Российский научный электронный журнал «Электронные библиотеки». – Москва, 2005. – http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2005/part2/Hohlov;
4. *Воройский Ф.С.* Электронные и традиционные библиотеки – суть не одно и тоже. // Российский научный электронный журнал «Электронные библиотеки». – Москва, 2003. – http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2003/part5/voroisky.

А.И.Панченко

ИНСТРУМЕНТАЛЬНОЕ СРЕДСТВО ДЛЯ АНАЛИЗА АССОЦИАТИВНЫХ СЕТЕЙ

Введение

Ассоциативный эксперимент – термин, утвердившийся в психологии для обозначения особого проективного метода исследования мотивации личности, предложен в начале XX в. К.Г. Юнгом и практически одновременно с ним М. Вертгеймером и Д. Кляйном. Испытуемый должен отвечать на определенный набор слов-стимулов как можно быстрее любым пришедшим ему в голову словом. Регистрируются тип возникающих ассоциаций, частота однотипных ассоциаций, латентные периоды (время между словом-стимулом и ответом испытуемого), поведенческие и физиологические реакции и др. По характеру этих данных можно судить о скрытых влечениях и "аффективных комплексах" испытуемого, его установках и т. п.

В самом простом виде результатом проведения ассоциативного эксперимента является набор ответов вида стимул → реакция. После проведения эксперимента на некотором множестве респондентов должна быть произведена обработка результатов, которые могут быть сведены *в Ассоциативный словарь*.

Существует множество ассоциативный словарей – это русский, английский, шведский, украинский, белорусский, латышский, киргизский и многие другие ассоциативные словари.

В статье словаря представляет собой лексему-стимул, которой сопоставляется список упорядоченных по частоте или по алфавиту (с указанием частоты) слов – реакций, полученных в психолингвистическом эксперименте.Первый «Словарь ассоциативных норм русского языка» (на материале 200 слов-стимулов) был издан в 1977 году.[РАС Ю.Н.Караулов]

Как показывают семантические и психолингвистические исследования, сфера ассоциативных отношений в лексике образует особую и достаточно устойчивую систему, которая и становится объектом описания в подобных словарях.

Работа, описанная далее в статье, в большой мере основана на статьях Русского Ассоциативного Словаря (РАС) поэтому немного расскажем про него. Вот как описывает словарь Ю.Н.Караулов, один из авторов РАС:«Русский ассоциативный словарь составляет первую и основную часть Ассоциативного тезауруса русского языка, который моделирует вербальную память и языковое сознание «усредненного» носителя русского языка.

Этот словарь является:— принципиально новым источником изучения языка и феномена владения языком;— базой для анализа путей и закономерностей формирования языкового сознания в онто- и филогенезе, формирования методологических и теоретических схем анализа языкового сознания;— базой для обучения русскому языку как родному или иностранному, а также средством оптимизации процессов речевого общения с человеком и ЭВМ (формирование сознания с помощью вербальных текстов в средствах массовой информации, речевое воздействие при интра- и интеркультурном общении).

Русский ассоциативный словарь, предназначенный для широкого круга потребителей,— первый словарь такого рода, поэтому структура и способ его составления нуждаются в объяснении.

Помимо РАС в основу программного изделия, описанного в статье, лег хорошо известный ассоциативный словарь английского языка (Kiss G.R., Armstrong C., Milroy R. «The Associative Thesaurus of English». Edinburgh: Univ. of Edinb.)

Постановка задачи

Таким образом, ассоциативные словари являются мощным инструментом как для исследований современного языка, так и для решения множества прикладных задач, к примеру данные этих словарей могут быть полезными людям творческих профессий, работающим с текстом. Однако, использование ассоциативных словарей это не частая практика, несмотря на всю явную пользу, которую они бы могли приносить. Причина этого, множество: ассоциативные словари издавались небольшими тиражами, некоторые из них, зачастую, трудно найти в продаже, а если они и есть, то достаточно дороги, бумажные версии словарей использовать не очень удобно. Многие издания ассоциативных словарей и вовсе являются библиографической редкостью.

Проблема доступности полезных знаний, содержащихся в ассоциативных словарях, стоит очень актуально. Вот почему возникла инициатива создания программного продукта для упрощения работы с данными словарей, причем через единый интерфейс для различных языков.

Кроме того, компьютерная версия словаря позволила бы построить модель *ассоциативной сети* языка и производить анализ и работу уже не только с множеством словарных статей, но и с целой сетью языка.

Если модель ассоциативной сети будет создана то станет возможным применение множества известных алгоритмов работы со структурами данных, к примеру различных методов поиска путей на графах.

Исходя из вышеуказанных предпосылок были поставлены следующие задачи по разработке инструмента:

1. Разработать единый формат для хранения данных ассоциативных экспериментов, т.к. на данный момент каждый из словарей хранится в своем собственном уникальном виде, что крайне затрудняет взаимную интеграцию и совместное использование результатов различных ассоциативных экспериментов.

2. Перевести имеющиеся базы словарей в данный формат.

3. Создать программный инструмент для работы с базами словарей. Приложение должно реализовывать поиск ассоциативных цепочек, в ассоциативных сетях русского и английского языков. При поиске следует учитывать *валентность* (кратность) ассоциативных связей. Сделать возможным разнообразные виды поиска – всех путей, только кратчайших и др. Этот программный продукт должен быть легко распространяемым, и удобным в применении, для того чтобы уйти от недостатков бумажных ассоциативных словарей.

Кроме того были поставлены некоторые первоочередные задачи по применению средства, после его создания:

1. Исследовать ассоциативную сеть на предмет топологии, структуры. Попытаться выделить ядро ассоциативной сети.

2. Исследовать, использую существующий инструмент, ассоциативный словарь русского и английского языков на предмет нахождения связности всех слов со всеми. Попробовать выяснить минимальное значение длины пути, которое бы приводило бы к любому слову из любого.

3. Исходя из всего вышесказанного можно резюмировать что была поставлена задача не только сделать более доступными данные ассоциативных словарей для конечных пользователей, но и привнести функциональность ранее недоступную, к примеру поиск *ассоциативных цепочек* в *ассоциативно-вербальной сети*. Система будет предназначаться для распространения как среди специалистов в области лингвистики и смежных областей, как и среди обычных пользователей заинтересованных в данных ассоциативных экспериментов.

Разрабатываемая система будет носить название «Ассоциативная Сеть» (далее возможно сокращение АС).

Аналоги системы «Ассоциативная сеть»

Несмотря на то что система Ассоциативная Сеть будет является достаточно узкоспециализированным программным продуктом, были найдено несколько аналогов имеющих подобную функциональность:

- Визуальный словарь (www.visualworld.ru, vslovar.org.ru).
- Словарь ассоциаций (www.slovesa.ru).
- RSO Semantic Network SDK – инструментарий разработчика.
- www.lexfn.com – интернет-ресурс, работающий с англоязычной ассоциативной сетью.

Визуальный словарь (www.visualworld.ru, vslovar.org.ru)

Проект VisualWorld.ru - это развитие проекта Визуальный словарь, представляющий собой поисковую информационную систему, объединяющую систему семантического поиска в Internet и поиск по энциклопедиям с визуализацией ассоциативных связей.

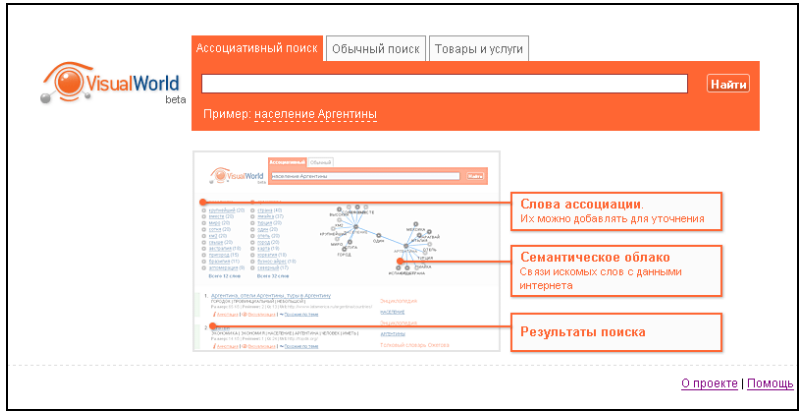


Рис.1. Главная страница

Основная особенность системы — поиск не точного вхождения слов запроса в страницу, а страниц, максимально точно описывающих то, что написано в запросе. Проект VisualWorld.ru - это система поиска знаний и ответов на вопросы. Для быстрого поиска есть "облегченная" поисковая форма viwo.ru.

Стоит отметить, что главной характерной особенностью данного проекта является визуализация результатов поиска в виде графа.

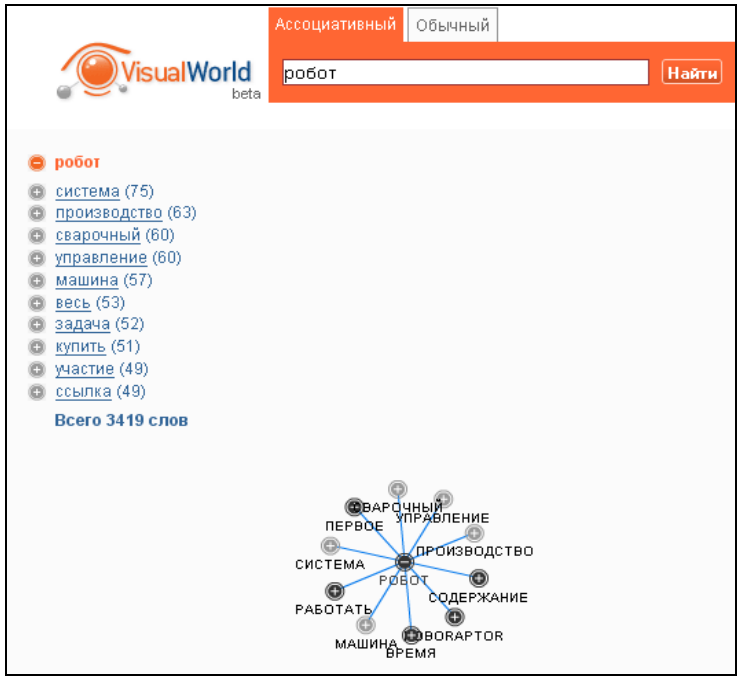


Рис.2. Пример поиска

Словарь ассоциаций (www.slovesa.ru)

Сотрудники компании «Витософт» Николай Бабичев и Виталий Ситницкий объявили об открытии бета-версии онлайнового словаря ассоциаций русского языка «Словеса.Ру» (www.slovesa.ru).

На момент открытия объем словаря насчитывал около 65000 слов. По словам Николая Бабичева, все ассоциативные связи построены автоматически на основе русскоязычных текстов, поэтому в словаре нет ни одной случайной ассоциации. Например, если по словарю «бобр» может быть «седым», значит где-то об этом написано, а чем больше в текстах упоминаний о том, что бобр седой, тем весомее слово «седой» в облаке свойств по запросу «бобр».

Разработчики полагают, что такой словарь будет полезен дизайнерам, художникам, другим творческим людям.

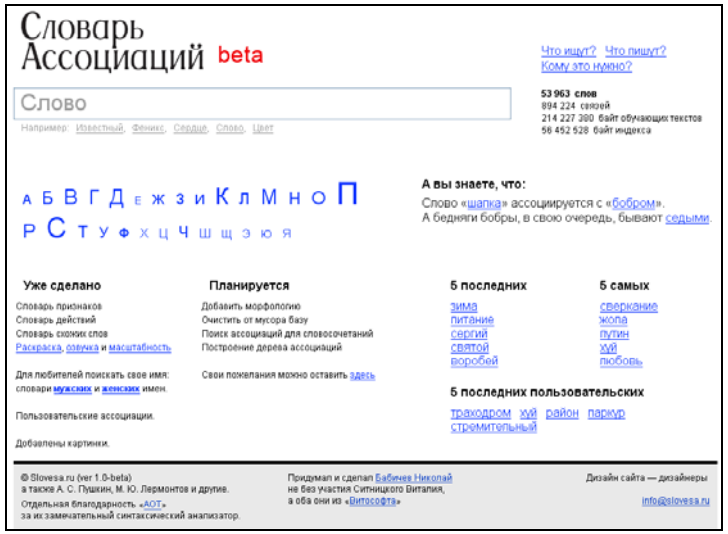


Рис.3. Главная страница

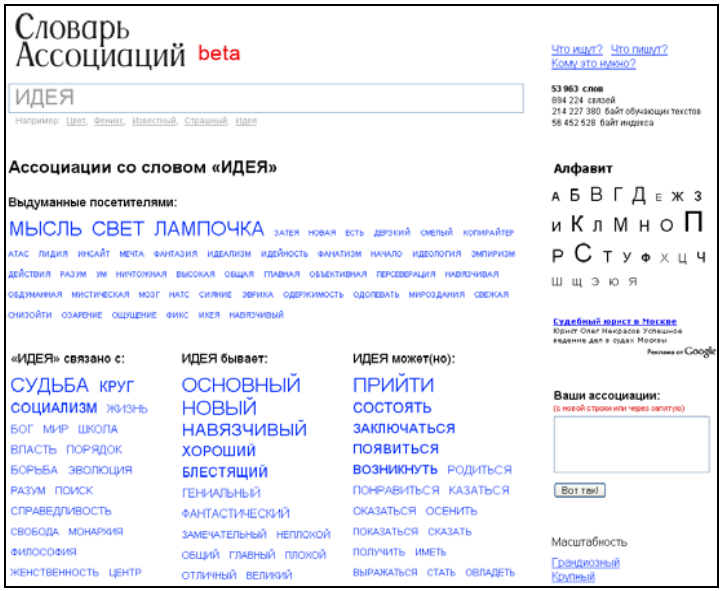


Рис.4. Пример поиска

RCO Semantic Network SDK

Средства библиотеки RCO Semantic Network позволяют автоматически анализировать содержание текстовых документов, представляя его в форме ассоциативной семантической сети. Ассоциативная семантическая сеть в программе представляет собой ориентированный граф, вершинами которого служат значимые темы, выделенные в анализируемом тексте, а дугами – связи между ними. С каждой вершиной связаны вес (значимость) и частота упоминания темы, а с каждой дугой – вес (сила) связи и частота подкрепления связи в тексте.

Помимо частоты упоминания в тексте, каждой теме присваивается вес от 1 до 100, отражающий ее значимость по отношению к другим темам. Пользователь может задать минимальный порог по весу, ниже которого темы не включаются в семантическую сеть.

Ассоциативные связи между темами выделяются на основе частоты их совместного появления в одном предложении. Пользователь может задать минимальный порог по частоте, ниже которого связи отбрасываются. В конечном представлении связь преобразуется в две противоположные по направленности дуги графа, которым присваиваются веса от 1 до 100, которые отражают условную вероятность упоминания первой темы совместно со второй – силу связи.

Дополнительно на каждую тему выдается тематический реферат, представляющий наиболее информативные фрагменты текста, в которых данная тема упоминалась. Общий реферат текста представляет компиляцию наиболее информативных фрагментов по ключевым темам. Подробность реферирования может настраиваться пользователем.

Lexfn (www.lexfn.com)

Интернет-ресурс предоставляет широкие возможности по поиску путей между знаками в ассоциативной сети. Система различает множество типов связей между словами, выполняет поиск не только непосредственно достижимых из начальной вершин сети, но и находит пути с ограничением по длине. На рисунке ниже изображена главная страница ресурса, а так же пример результата запроса.

Среди сильных сторон системы можно выделить хорошую проработку ассоциативной сети, множество различных типов связей, объемную базу данных и высокую скорость выполнения запросов.

В системе отсутствует тезаурус и, как следствие, нет полноценной реализации пассивного режима работы когнайзера.

Рис.5. Главная страница

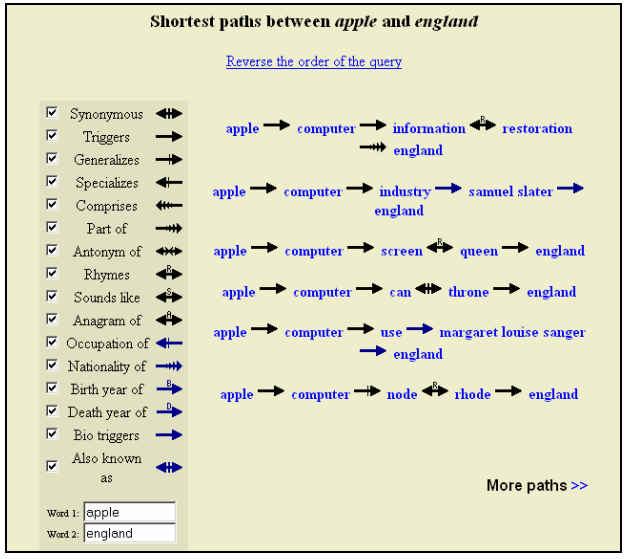


Рис.6. Результаты запроса

Особенностью интернет-проекта в том, что он не доступен локально, то есть при отсутствии подключения к сети Интернет. Пополнением базы данных занимаются разработчики, вносить изменения самостоятельно не позволяется. Система работает только в рамках английского языка, русскоязычного варианта не существует.

Результаты сравнения аналогов

Несмотря на то что существуют аналоги системы, которую планируется создать, тем не менее ни одно из реализованных решений не обладает теми возможностями которые должны быть реализованы в программном продукте Ассоциативная Сеть. К примеру, нигде не поддерживается сразу поиск по нескольким ассоциативным словарям для разных языков, так же ни одна из систем не предоставляет возможность поиска ассоциативных цепочек в ассоциативной сети. Помимо всего этого ни одна из программ не предоставляет пользователю информацию из Русского Ассоциативного Словаря.

Исходя из всего вышесказанного, можно сделать вывод что разработка программного продукта Ассоциативная Сеть является целесообразной и не будет повторять уже реализованное решение в данной области.

Модель ассоциативно-вербальной сети (АВС)

Результаты ассоциативного эксперимента можно сохранить в виде структуры данных для того чтобы иметь возможность применять известные хорошо проработанные алгоритмы для решения прикладных задач компьютерной лингвистики.

Подходящим математическим аппаратом для представления данных ассоциативного эксперимента является теория графов. Представляется удобным моделирование ассоциативно-вербальной сети в виде ориентированного взвешенного графа содержащего циклы.

Таким образом, можно ввести формальное определение а*ссоциативной сети*.

Ассоциативной сетью будем называть ориентированный взвешенный граф $Anet = (Dic, Asn)$ с весовой функцией $w \to R$.

Dic – множество вершин графа, причем $Dic = Stim \cup React$, где

$Stim$ – множество слов-стимулов в словаре ассоциативного эксперимента, $React$ – множество слов-реакций в словаре ассоциативного экс-

перимента. Множество Dic будем называть *объединенным словарем ассоциативного эксперимента*.

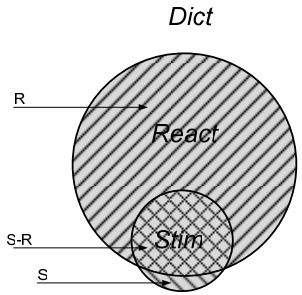


Рис.7. Объединенный словарь ассоциативного эксперимента

Asn – множество дуг графа, причем такое что $Asn \subseteq Dic \times Dic$, из этого следует, что в ассоциативной сети возможны *циклические дуги*. Для каждой дуги из множества Asn весовая функция $\omega \rightarrow R$ задает значение метки дуги, имеющее значение валентности (частоты) ассоциативной связи между двумя вербальными единицами из *объединенного словаря ассоциативного эксперимента*.

К описанной выше модели можно применить любой из множества алгоритмов, известных в дискретной математике, к примеру, по нахождению кратчайших путей или по выделению остовного дерева минимального веса, поэтому представляется достаточно удобным моделирование ассоциативно-вербальной сети в виде ориентированного графа.

Построим фрагмент ассоциативной сети, взяв за основу данные Русского Ассоциативного словаря [7:

$$Anet = (Dic, Asn)$$

Dic = {Слово, Фраза, Речь, Дело, Говорить, Разговор, Номер, Жизнь, Время, Телефон, Один, Деньги}

$$Asn = \{a_1, a_2, a_3, \ldots, a_{18}\}$$

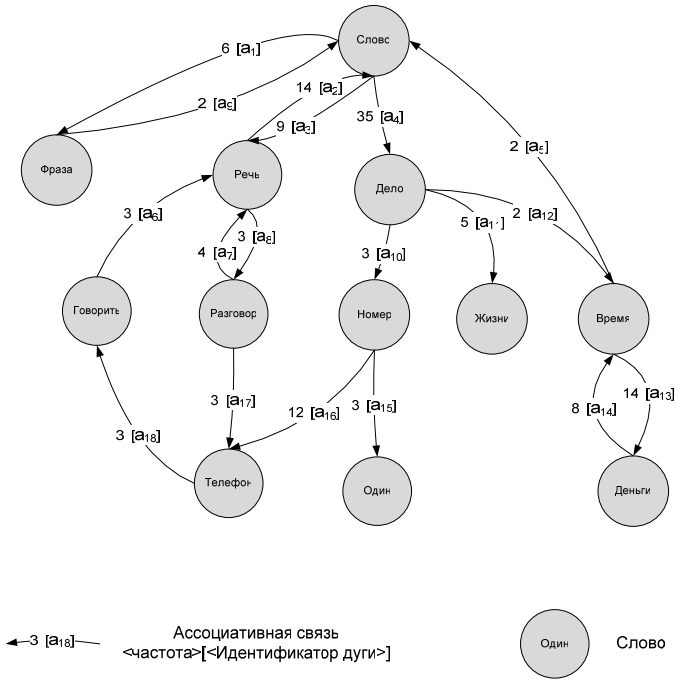


Рис.8. Фрагмент ассоциативно-вербальной сети русского языка (ориентированный граф)

Введем формальное понятие ассоциативной цепочки, для этого определим понятие ассоциативной связи.

Ассоциативная связь – дуга графа ассоциативной сети, $a = \{d_s, d_r\} : d_s \rightarrow d_r$, где $d_s, d_r \in Dict$; $d_s \in Stim$; $d_r \in React$.

Ассоциативной цепочкой в ассоциативной сети будем называть конечное множество, мощностью k: $Chn = \{a_1, a_2, \ldots, a_k\}$, $Chn \subseteq Asn$. Причем для $\forall a_i \rightarrow a_{i+1}$, если $\exists a_{i+1}$.

Точность модели ассоциативно-вербальной сети и её адекватность и применимость напрямую зависит от точности данных ассоциативного эксперимента.

Исходя из данных Английского ассоциативного эксперимента [333], насчитывается около 23тыс. слов и около 113тыс. ассоциативных связей.

Граф ассоциативно-вербальной сети русского языка, по данным Русского Ассоциативного Словаря [8] состоит из более чем 32тыс. слов (вершин графа) и около 325тыс. ассоциаций (дуг графа).

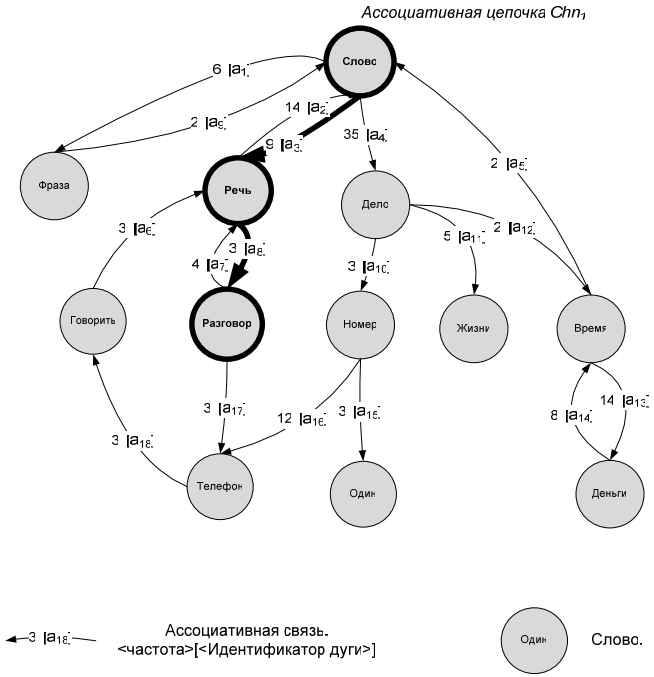


Рис.9. Пример ассоциативной цепочки на фрагменте ассоциативно-вербальной сети.

Особенность проведения ассоциативного эксперимента состоит в том, что множество реакций не являются стимулами, причем это характерно для ассоциативных связей с единичной валентностью. Из-за этого свойства граф, построенный по результатам ассоциативного эксперимента, имеет характерную структуру, которая видна из рисунка 10, а вершины *ассоциативной сети* можно классифицировать.

Листовой вершиной будем называть такое слово $d \in React$, для которого $\exists a \in Asn : a = \{d, a_i\}, a_i \subseteq React$. Иначе говоря, d называется *листовой вершиной*, если оно является только реакцией, но не стимулом для какой-либо ассоциативной связи.

Корневой вершиной будем называть такое слово $d \in Stim$, для которого $\nexists a \in Asn : a = \{a_i, d\}, a_i \subseteq Stim$. Иначе говоря, d называется *корневой вершиной*, если оно является только стимулом, но не реакцией для какой-либо ассоциативной связи.

Основной вершиной будем называть такое слово $d \in Dict$, для которого
$\exists a_{stim} \in Asn : a = \{d, a_i\}, a_i \subseteq React$ *и также* $\exists a_{react} \in Asn : a = \{a_i, d\}, a_i \subseteq Stim$

Таким образом $d \subseteq (Stim \cap React)$. Иначе говоря d называется *основной вершиной*, если оно является одновременно и стимулом и реакцией.

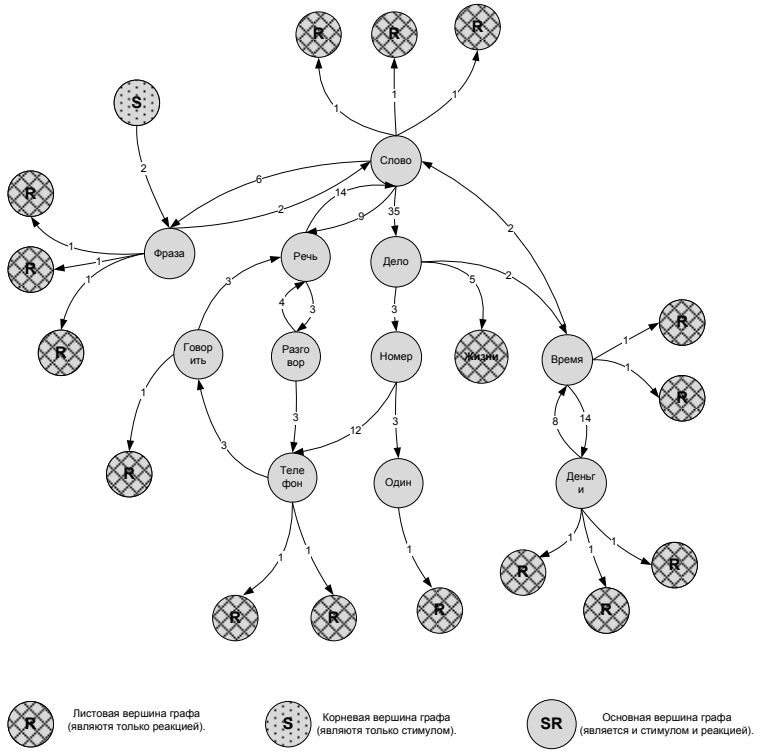


Рис. 10. Классифицированный граф ассоциативной сети

Анализ алгоритмов поиска ассоциативных цепочек в ассоциативно-вербальной сети

Мы рассмотрим четыре алгоритма для поиска цепочек в ассоциативно-вербальной сети, причем реально два из них предназначены для поиска не в сети а в древовидной структуре (алгоритм поиска в ширину и в глубину), а другие два предназначены для поиска уже в сети (алгоритм Дейкстры и Беллмана-Форда).

С первого взгляда несколько странно сравнивать работу алгоритмов на разных структурах данных. Тем не менее, это становится возможным т.к. ассоциативно-вербальную сеть можно привести к дереву во время поиска кратчайших ассоциативных цепочек. Пример приведения представлен на рис. 11–12.

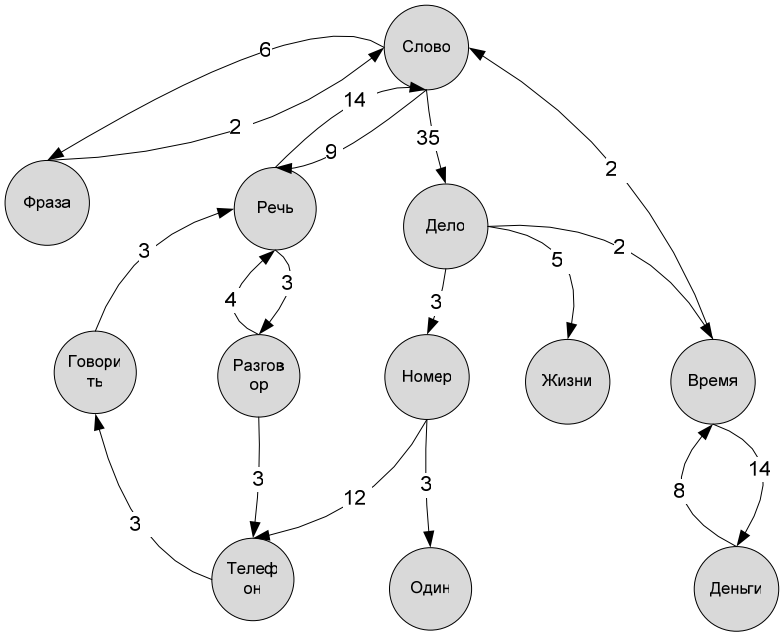


Рис.11. Исходный граф ассоциативной сети

При каждой итерации поискового алгоритма, проверяющей список смежности очередной вершины делаем проверку на содержание каждой вершины во множестве осмотренных. Если вершина содержится, то

связь отбрасывается. Таким образом получаем древовидную структуру из сетевой. Это продемонстрировано на рис. 12.

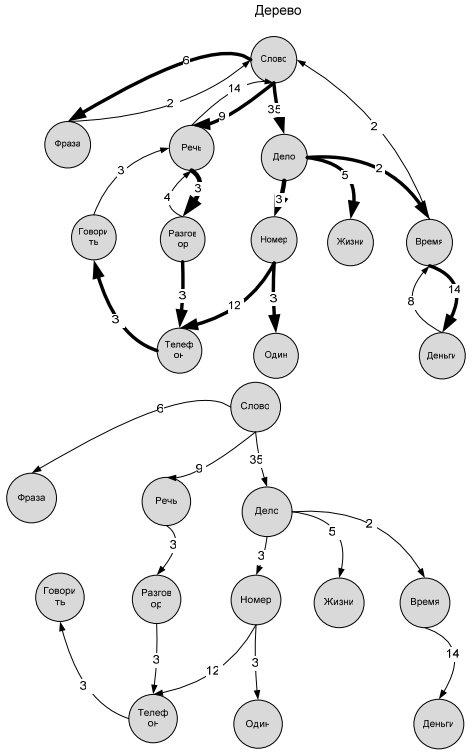


Рис.12. Процесс преобразования к древовидной структуре

Алгоритм поиска в глубину

Краткое описание. Поиск в глубину - вероятно, наиболее важная ввиду многочисленности приложений стратегия обхода графа. Идея этого метода — идти вперед в неисследованную область, пока это возможно, если же вокруг все исследовано, отступить на шаг назад и искать новые возможности для продвижения вперед.

Главное отличие от поиска в ширину состоит в том, что при поиске в глубину в качестве активной выбирается та из открытых вершин, которая была посещена последней. Для реализации такого правила выбора наиболее удобной структурой хранения множества открытых вершин является стек: открываемые вершины складываются в стек в том поряд-

ке, в каком они открываются, а в качестве активной выбирается последняя вершина.

Алгоритмическая сложность. Время поиска в глубину линейно зависит от размера представления графа G.

T = O (V+E), где: V-количество дуг, E-количество вершин.

Стоит отметить что $F \blacksquare n^H$, где n – средняя степень исхода для узла ассоциативно-вербальной сети, H – длина искомого пути.

Алгоритм поиска в ширину

Краткое описание. Поиск в ширину — метод обхода и разметки вершин графа. Поиск в ширину выполняется в следующем порядке: началу обхода S приписывается метка 0, смежным с ней вершинам — метка 1. Затем поочередно рассматривается окружение всех вершин с метками 1, и каждой из входящих в эти окружения вершин приписываем метку 2 и т. д.

Основная особенность поиска в ширину, отличающая его от других способов обхода графов, состоит в том, что в качестве активной вершины выбирается та из открытых, которая была посещена раньше других. Именно этим обеспечивается главное свойство поиска в ширину: чем ближе вершина к старту, тем раньше она будет посещена. Для реализации такого правила выбора активной вершины удобно использовать для хранения множества открытых вершин очередь - когда новая вершина становится открытой, она добавляется в конец очереди, а активная выбирается в ее начале.

Алгоритмическая сложность. Время поиска в ширину линейно зависит от размера представления графа G.

T = O(V+E), где: V-количество дуг, E-количество вершин.

Алгоритм Беллмана-Форда

Краткое описание. Алгоритм Беллмана- Форда — алгоритм поиска кратчайшего пути во взвешенном графе. В отличие от алгоритма Дейкстры, алгоритм Беллмана-Форда допускает рёбра с отрицательным весом.

В основе алгоритма Форда-Беллмана лежит метод динамического программирования. Для заданного взвешенного ориентированного графа G=(V,E) с истоком S и весовой функцией W алгоритм возвращает логическое значение, указывающее на то, содержится ли в графе цикл с отрицательным весом, достижимый из истока. Если такой цикл сущест-

вует, в алгоритме указывается, что решения не существует. Если же циклов нет, алгоритм выдает кратчайшие пути и вес.

Алгоритмическая сложность. За время T = O(V*E) алгоритм находит кратчайшие пути от одной вершины графа до всех остальных, где V—количество дуг, E—количество вершин.

Алгоритм Дейкстры

Краткое описание: Алгоритм Дейкстры решает задачу о кратчайшем пути из одной вершины во взвешенном ориентированном графе G=(V,E) в том случае, когда веса ребер неотрицательны. При хорошей реализации алгоритм Дейкстры более производителен, чем алгоритм Беллмана-Форда.

В алгоритме Дейкстры поддерживается множество вершин S, для которых уже вычислены окончательные веса кратчайших путей к ним из истока s. В этом алгоритме поочередно выбирается вершина $u \in \{V - S\}$, которой на данном этапе соответствует минимальная оценка кратчайшего пути. После добавления этой вершины u в множество S производится ослабление всех исходящих из нее ребер. В приведенной ниже реализации используется неубывающая очередь с приоритетами Q, состоящая из вершин, в роли ключей для которых выступает значения d.

Алгоритм Дейкстры называется «жадным» т.к. из множества $\{V - S\}$ для помещения в множество S всегда выбирается самая «легкая» или «близкая» вершина.

Алгоритмическая сложность. Время работы алгоритма Дейкстры, в общем случае, квадратично зависит от количества вершин графа и равно $T = O(V^2 + E)$. При соблюдении определенных условий время работы алгоритма может достигать значения T = O(V·lgV + E), где V—количество дуг, E—количество вершин.

Волновой алгоритм

Выберем для реализации алгоритма поиска ассоциативных цепочек в ассоциативной сети алгоритм волнового фронта. Он обладает временем поиска путей линейно зависимым от количества вершин графа:

T = O(V+E), где: V-количество дуг, E-количество вершин.

Этот алгоритм является легким в реализации и вполне подойдет для создания тестовой версии системы Ассоциативная Сеть.

Стоит отметить, что количество вершин графа равно $V = n^H$, где n – средняя степень исхода вершины графа, H – максимальная длина

искомой ассоциативной цепочки. Таким образом, время поиска будет зависеть от длины искомого пути по следующей зависимости:

$$O(V + E) \cong O(n^n).$$

Выбор архитектуры для программного продукта Ассоциативная Сеть

На сегодняшний день существует множество вариантов архитектуры и тем более технологий, реализующих эти архитектуры. Нам следует выбрать наиболее подходящий для нас вариант.

Напомню что при формулировании задач, которые ставились до написания программы, требовалось чтобы программный продукт был легко распространяемым и удобным в применении, для того чтобы уйти от недостатков бумажных ассоциативных словарей. Кроме этого не стоит забывать о том, что важной конкурентной особенностью программы должна являться возможность поддержки множества словарей в едином формате.

Таким образом, можно сформулировать требования к архитектуре:

1. *Прозрачность установки.* Возможность легкого развертывания на компьютере пользователя
2. *Автономность работы.* Поддержка режима работы без подключения к интернету. Это необходимо для небольшого количества пользователей, которые имеют права хранить базы данных с ассоциативными словарями у себя на компьютере, они имеют возможность редактировать эти базы.
3. *Инкапсуляция данных.* Возможность не распространять базы данных с ассоциативными сетями конечным пользователям (ассоциативные словари часто являются интеллектуальной собственностью и открытый доступ к ним крайне не желателен).
4. *Модульность архитектуры.* Возможность легко отделить визуальное представление от логики работы программы. Это нужно для того чтобы в дальнейшем не привязываться к конкретной платформе, к примеру Windows-приложения, а сделать возможным безболезненный переход на другую архитектуру (Web-приложение, Web-сервис и т.п.).
5. *Высокая скорость работы.* Это необходимо для обеспечения интерактивности работы т.к. предполагается реализовывать ресурсоемкие алгоритмы поиска на структурах данных, а для этого необходимы гибкие возможности по оптимизации скорости исполнения кода.

6. *Масштабируемость*. Возможность работы большого количества пользователей без радикальных изменений в архитектуре.

Приведем перечень наиболее распространенных архитектур:

*Файл-серверная архитектура*_Модель "файл-сервер" - архитектура вычислительной сети типа "клиент-сервер", в которой сервер предоставляет в коллективное пользование дисковое пространство, систему обслуживания файлов и периферийные устройства.

Примеры типовой реализации данной архитектуры – в качестве сервера выступает СУБД Access или Paradox, клиент Win32 приложение.

Клиент-серверная архитектура (сервер базы данных). Модель "сервер базы данных" - архитектура вычислительной сети типа "клиент-сервер", в которой пользовательский интерфейс и логика приложений сосредоточены на машине-клиенте, а информационные функции (функции СУБД) - на сервере. Обычно клиентский процесс посылает запрос серверу на языке SQL.

Классическим примером реализации данной архитектуры является СУБД Oracle/Informix/SQL Server в качестве сервера и Windows/*nix клиент.

Сервер приложений (Web-приложение). Модель "сервер приложений" - архитектура вычислительной сети типа "клиент-сервер", в которой функциональная логика размещена на сервере, а на машине-клиенте выполняется только компонент представления.

Типичным примером реализации данной архитектуры является любое web-приложение. Веб-приложение обладает уникальной кросс-платформенностью и прозрачностью доступа, т.к. работать с программой, реализованной с помощью этой технологии, можно используя любую операционную систему и платформу, главное это наличие доступа в интернет у клиента.

Стоит отметить две различных реализации – используя компилируемые языки и используя интерпретируемые языки программирования. К первой группе относятся такие технологии как PHP, ASP, Ruby и другие. Преимуществом данных технологий является относительная легкость освоения языков, простота развертывания. Недостатком же является то, что при каждом вызове странички необходима интерпретация кода. Плохая модульность приложения написанного используя данные технологии так же может быть серьезной помехой.

К другой группе можно отнести технологии, использующие прекомпилированные сборки при работе сайта (Java Beans и .NET Assemblies). К этой группе можно отнести платформу J2EE от Sun и технологию ASP.NET от Microsoft. Скорость работы приложений написанных

на данных платформах как правило выше, и что не маловажно эти технологии предоставляют очень удобно реализовать модульную объектно-ориентированную архитектуру приложения. Богатейший доступ к различным библиотекам классов тоже является весомым преимуществом.

Рассмотрим различные варианты архитектур:

Требования	1	2	3	4	5	6	Итого
α	0.1	0.1	0.2	0.1	0.3	0.05	1
Файл-серверная	0.7	1	0	0.5	0.5	0.3	0.4
Клиент-серверная (сервер базы данных)	0.2	1	1	1	1	1	0.92
Сервер приложений (интерпретируемый язык)	1	0	1	0.3	0.5	1	0.68
Сервер приложений (компилируемый язык)	1	0	1	1	1	1	0.9

α – коэффициент важности фактора, * – обязательный параметр

Таким образом, видим, что наиболее подходят для наших целей *клиент-серверная архитектура* и *архитектура тонкого клиента с компилируемым языком*. Значение итоговой оценки практически одинаково для этих двух вариантов. Но не стоит забывать о том что необходимо обеспечение возможности автономной работы приложения. Клиент-серверная архитектура позволяет реализовать это свойство, хотя бы и ценой повышенной сложности при установке. Существуют облегченные и бесплатные версии клиент-серверных СУБД, которые можно развернуть у пользователя, например SQL Server Express Edition . Развертывание же веб-сайта на машине пользователя вызывает куда большие проблемы т.к. помимо сервера БД еще необходимо установить и веб-сервер, а так же его правильно настроить, для того чтобы программа заработала. Безусловно, можно сделать программу инсталлятор, которая бы производила все эти вспомогательные действия, однако она будет неоправданно сложна. В варианте с тонким клиентом возникает еще одна проблема – веб-сервер обладает меньшей транспорентностью к установке на различные ОС, к примеру, Internet Information Services невозможно установить на ОС Windows XP Home Edition, крайне популярную сегодня.

Исходя из всего вышесказанного, останавливаем свой выбор на *клиент-серверной* архитектуре.

Стоит отменить, что параметры *автономность работы* программы и *инкапсуляция данных* являются взаимно-исключающими, потому как передача конечному пользователю базы ассоциативного словаря является неприемлемой для большинства случаев. Несмотря на это, нужно обеспечить возможность работы без подключения к интернету.

Эту неоднозначность было решено преодолеть, введя два режима работы программы – локальный и сетевой. В локальном режиме работы программы приложение работает с сервером СУБД, установленным на компьютере привилегированного пользователя, в сетевом же режиме, клиент обращается к удаленному серверу базы данных, расположенному в интернете (сервер должен быть сконфигурирован таким образом, чтобы можно было бы только вызывать определенные хранимые процедуры, но не делать выборки из таблиц).

Выбор технологий реализации архитектуры

Как было показано выше наиболее оптимальной архитектурой для программного продукта «Ассоциативная сеть» является клиент-серверная архитектура приложения, в которой прикладные и пользовательские сервисы реализованы на клиентской рабочей станции, а данные централизованно хранятся на сервере. В этой модели клиенты подключаются непосредственно к серверу, на все время работы приложения.

Таким образом, следует выбрать те конкретные платформы, которые будут реализовывать обе части.

Выбор платформы для клиента

Первым делом стоит отметить, что исходя из современных реалий, подавляющее большинство пользователей работают с операционными системами семейства Microsoft Windows, так что будем считать, что большинство потенциальных пользователей программы будет использовать платформу Windows. Тем не менее, возможность относительно легкой реализации кросс-платформенности, безусловно, является плюсом.

Отметим требования которые будут играть главную роль в выборе платформы, на которой планируется реализовать продукт Ассоциативная Сеть:

1. *Легкость установки.* Для клиентской части приложения важна максимальная легкость и прозрачность установки на компьютер

пользователя. Если процесс развертывания приложения на машине клиента будет громоздким, долгим и требующим большого количества ресурсов, то тогда человек использующий программу может решить что выгода от использования продукта не настолько велика, насколько велики те сложности связанные с накладными расходами по установке ПО.

2. *Скорость и удобство написания.* Для каждой конкретной технологии и языка программирования существуют свои *средства разработки.* Даже если алгоритмический язык обладает всеми качествами, которые требуются (к примеру, поддержка объектно-ориентированности), средства разработки программ на этом языке могут быть не слишком удобными. Проблему, так же может вызвать нехватка стандартных библиотек, необходимых для легкого написания программы, или необходимость покупки лицензии на использование данных библиотек. Иначе говоря, данный параметр комплексно оценивает наличие для технологии удобных средств для быстрой разработки приложения (Rapid Application Development – RAD). В качестве примера можно привести тот факт что для языков семейства .NET и Java существует множество средств автоматизированного *рефакторинга* кода, в отличие от C++.
3. *Модульность.* Легкость интеграции и повторного использования уже написанного кода с новым. Современные программы составляют таким образом чтобы разделить уровни абстракции на разные логические уровни – графическое представление, логика работы, логика доступа к данным и др. Это становиться возможным используя объектно-ориентированную технологию.
4. *Скорость работы.* Как минимум некоторая часть логики работы поискового алгоритма будет исполнятся на клиентской стороне, таким образом скорость исполнения этих алгоритмов является важным критерием при выборе технологии.
5. *Затраты на изучение.* Этот параметр характеризует степень дополнительных инвестиций в человеческий капитал для того чтобы эффективно реализовывать продукт используя данную технологию. Этот параметр немаловажен т.к. скорость и качество разработки напрямую зависит от того насколько освоены исполнителем те или иные инструменты. Подобрав даже самые оптимальные и современные инструменты можно, без должного опыты их применения создать менее качественный продукт, чем если применять более хорошо освоенные техники.

6. *Кроссплатформенность.* Возможность запуска клиентского приложения на разных операционных системах. Как было указано выше, основная масса людей заинтересованных в использовании программного продукта Ассоциативная Сеть используют ОС Windows, как наиболее распространенную сегодня, поэтому важность этого параметра достаточно низкая.

Итак, приведем перечень тех параметров, которые являются наиболее важными при выборе платформы для клиента:

Требования	1	2	3	4	5	6	Итого
α	0.1	0.20	0.3	0.2	0.15	0.05	
Java	0.5	1	1	0.6	0.6	1	0.80
.NET Framework	0.5	1	1	0.6	1	0.5	0.85
Win32 API (C/C++)	0.9	0.1	0.2	1	0.4	0	0.43
Borland C++ Bulder/Delphi	0.9	0.5	0.5	0.9	1	0	0.67

Выбор платформы для сервера (СУБД)

Программный продукт Ассоциативная Сеть должен поддерживать одновременно множество ассоциативных словарей. Результаты ассоциативного эксперимента могут быть достаточно объемными, т.к. в результате эксперимента опрашиваются десятки тысяч респондентов, каждому из которых задают множество вопросов. Операции поиска ассоциативных цепочек могут требовать многократного обращения к серверу и вызывать большую нагрузку на СУБД. Вот почему база данных должна иметь возможность хранить большие объемы данных и быть производительной, иначе продукт Ассоциативная Сеть не будет иметь конкурентных преимуществ над своими аналогами.

Базы данных с результатами ассоциативных экспериментов изначально были представлены в формате СУБД Paradox. Данная СУБД не является клиент-серверной и обладает низкой производительностью, поэтому было решено перенести данных из нее в более подходящую СУБД для программы Ассоциативная Сеть.

Приведем краткое описание самых распространенных систем управления базами данных:

Informix — СУБД класса Enterprise (Предприятие), подходящая для управления данными в среднем и крупном бизнесе.

Отличается высокой надёжностью и быстродействием, встроенными средствами восстановления после отказов, наличием средств реплика-

ции данных и обеспечения высокой доступности, возможностью создания распределённых систем.

Поддерживаются почти все известные серверные платформы: IBM AIX, GNU/Linux (RISC and i86), HP UX, SGI Irix, Solaris, Windows NT (NT, 2000).

В линейку программных продуктов под общим названием "Informix" входят множество СУБД: Informix Dynamic Server Enterprise Edition, Informix Extended Parallel Server, Informix Dynamic Server Express, Informix Red Brick Warehouse и др.

Среди них присутствуют СУБД, пригодные для развертывания на локальной машине (Informix Standard Engine).

Oracle Database или Oracle RDBMS — объектно-реляционная система управления базами данных (СУБД). Поддерживает практически все программно-аппаратные платформы: Linux, Microsoft Windows, Solaris, AIX5L, HP-UX, PA-RISC, IBM z/OS, Mac OS X Server и др. Содержит множество различных редакций: Enterprise Edition, Standard Edition, Standard Edition One, Personal Edition, Lite и др. Существует версия Express Edition, распространяемая бесплатно.

Возможности и производительность систем управления базами данных Oracle гораздо выше чем у большинства конкурентов. Тем не менее, данная СУБД обладает самой высокой стоимостью лицензии (достигающей десятков тысяч долларов), и так же требует большого количества администрирования, для эффективной работы.

Применяется, в основном в крупных распределенных системах с высокой нагрузкой.

MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

MySQL имеет API для языков C, C++, Эйфель, Java, Perl, PHP, Python, Ruby, Smalltalk и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC.

В актив этой СУБД можно зачислить ее большую распространенность – так на большинстве как windows, так и *nix веб-хостингов есть поддержка MySQL.

Microsoft SQL Server. Реляционная система управления базами данных (СУБД), используется для небольших и средних и крупных баз данных.

Данная СУБД очень тесно интегрирована с платформой Microsoft .NET и средствами разработки для данной платформы. Так, к примеру, хранимые процедуры можно писать не только на языке SQL, но и так же на языках C# и VB.NET.

Имеется версия Microsoft SQL Server Express являющаяся свободно распространяемой версией SQL Server, походящей для легкого развертывания у клиента и дальнейшей автономной работы.

Было принято решение об использовании СУБД SQL Server в качестве сервера базы данных для программы Ассоциативная Сеть потому что, производительность этого сервера позволит работать с большими объемами данных и подключить множество словарей. Наличие версии SQL Server Express дает возможность легкого развертывания на машине клиента для автономной работы приложения без подключения к интернету. Распространенность данной СУБД позволит в дальнейшем перейти к архитектуре Веб-приложения (практически все веб-хостинги на базе Windows поддерживают SQL Server).

Для реализации программного продукта Ассоциативная сеть была выбрана клиент-серверная архитектура на основе сервера базы данных. В качестве сервера базы данных выбрана СУБД Microsoft SQL Server 2005, а в качестве технологии для реализации клиентской части приложения выбрана платформа Microsoft .NET Framework.

Описание созданного программного продукта «Ассоциативная Сеть»

Обзор программы

С учетом требований и проектных решений, которые были описаны выше, была реализована windows-версия программы «Ассоциативная Сеть». Мы опускаем детали имплементации, потому что они заслуживают отдельного рассмотрения, особенно разработка единого формата хранения данных ассоциативных экспериментов.

С помощью созданной программы можно производить поиск ассоциативных цепочек в АВС двух языков: русского и английского. При этом, возможно задавать различные параметры поиска, для того чтобы найти именно то что необходимо.

Помимо функции *поиска цепочек*, являющейся основной задачей программы, реализованы возможности *поиска всех реакций на заданный стимул* для русского и английского словаря, а так же *функция построения фрагмента ассоциативно-вербальной сети*.

Таким образом, программа решает три задачи по работе с АВС: поиск ассоциативных цепочек, построение фрагмента ассоциативной сети и нахождение множества реакций для заданного стимула. В следующем разделе мы рассмотрим более подробно каждую из этих возможностей.

Созданная программа имеет возможность работать в двух режимах – *удаленном* и *локальном*. При работе в *удаленном* режиме для запуска программы необходим доступ компьютера в Интернет. Для начала работы в этом режиме достаточно скопировать файлы программы на жесткий диск и запустить само приложение. Когда вы запустите приложение соединиться с базой данных ассоциативных экспериментов, находящейся на интернет сервере и будет работать с ней. Для комфортной работы не нужно высокой скорости подключения к интернету, это делает возможным работу программы даже через dial-up модем или GPRS.

При работе в *локальном* режиме программа может работать без подключения к интернету – она будет подключаться к серверу базы данных, установленному на том же компьютере что и программа, или в той же локальной сети, что и компьютер пользователя. Функциональность программы в данном режиме никак не меняется. Стоит отметить тот факт, что удаленная версия программы, которая работает через интернет, свободно распространяется и доступна для загрузки на официальном сайте проекта Ассоциативная Сеть [898]. В то же время локальная версия, содержащая помимо самой программы все базы данных ассоциативных экспериментов и некоторые дополнительные файлы для работы в локальном режиме, имеет ограничения на распространение из-за авторских прав на базы данных.

Локальная версия программы, помимо автономной работы может работать и в удаленном режиме. Во время работы программы можно переключаться между локальным и Интернет режимом работы с помощью меню в нижнем левом углу основного окна программы.

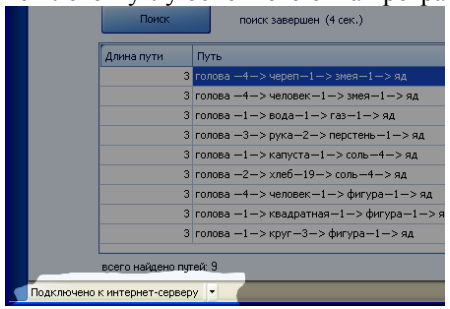


Рис.13. Изменение режима работы программы

Установка программы и системные требования

Для установки программы нужно выполнить несколько несложных действий:

1. Установить пакет Microsoft .NET Framework 2.0. Он является бесплатным и его можно загрузить с сайта Microsoft [000].
2. Загрузить с архив с программой Ассоциативная Сеть с официального сайта проекта (http://www.associative.ru/).
3. Распаковать содержимое архива в любое место на жестком диске.

Программа работает практически со всеми современными операционными системами семейства Microsoft Windows: Windows XP, Windows 2000, Windows 98, Windows ME, Windows Server 2003, Windows Vista.

Для установки пакета .NET Framework потребуется 280 Мб свободного места на жестком диске. Сама же программа «Ассоциативная Сеть» занимает на жестком диск объем не более 5 Мб.

Для функционирования программы необходим доступ в интернет (обязательно должен быть открыт TCP порт 1433). Желателен монитор с разрешением не менее 800×600.

Поиск ассоциативных цепочек

Программа позволяет искать по заданному слову-стимулу и слову-реакции все множество ассоциативных цепочек (путей) имеющихся в АВС для указанной пары.

Для того чтобы начать поиск путей достаточно просто запустить программу (файл AssociativeNet.exe). Раздел по поиску цепочек установлен по умолчанию, поэтому никаких дополнительных действий производить не потребуется.

Основное окно программы

Рассмотрим составляющие основного окна (рис. 13.) более подробно. Для того чтобы найти ассоциативные цепочки следует ввести стимул и реакцию в поля с соответствующими названиями и нажать кнопку «Поиск». После этого программа начнет поиск путей. В зависимости от заданных параметров и введенных слов он может занять от одной секунды до минуты и более. В любой момент можно отменить поиск цепочек для заданных слов и изменить условия запроса:

После завершения поиска на экран будет выведен результат в виде множества найденных цепочек, к примеру: **«война —7→ кровь—5→ любовь»**, для пары «война, любовь». Цифрой обозначена частота ассоциативной связи между словами, так, в указанной выше цепочке, связь **«война —7→ кровь»** имеет частоту равную семи. Подобная запись оз-

начает, что семь человек в ассоциативном эксперименте ответили на стимул «война» словом «кровь».

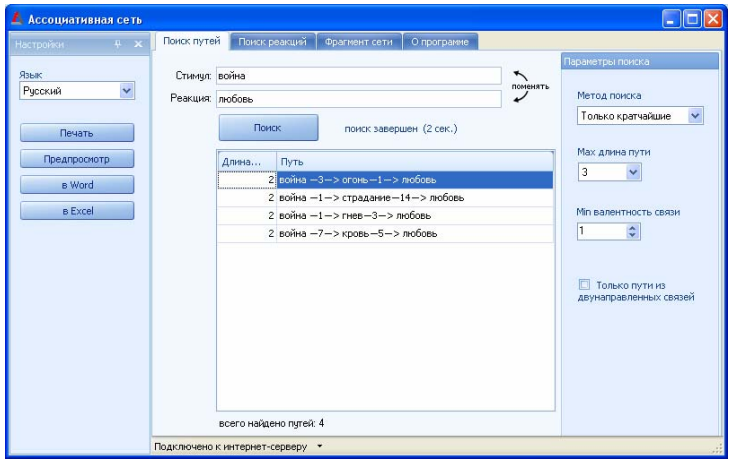


Рис.14. Основное окно приложения «Ассоциативная Сеть», раздел «Поиск путей»

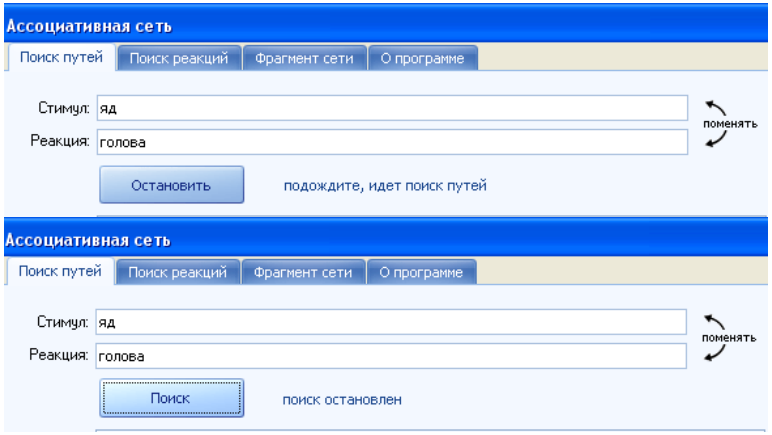
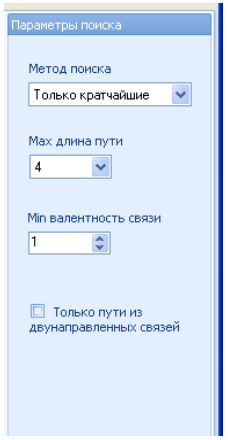


Рис.15. Остановка поиска ассоциативных цепочек

В программе можно задавать несколько параметров поиска. По умолчанию отыскиваются только кратчайшие пути с максимальной

длиной пути равной четырем, минимальная частота при этом равна единице.

Для задания параметров следует обратиться к панели «Параметры поиска» в левой части окна. Разберем значение каждого из параметров поиска.



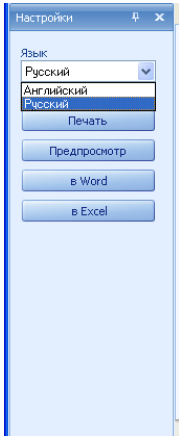


Рис.16. Параметры поиска. Рис.17. Панель выбора языка

Метод поиска

Возможны два метода поиска цепочек в ассоциативной сети:

«Только кратчайшие». При задании этого варианта ищутся кратчайшие пути, в пределах заданной максимальной длины пути (параметр «Максимальная длина пути»). Когда находится хотя бы одна цепочка поиск прекращается. Данный метод поиска является гораздо более быстрым по сравнению с поиском всех путей.

«Все пути». Данный метод отыскивает все возможные ассоциативные цепочки, в пределах максимальной длины пути (параметр «Максимальная длина пути»). Время работы поиска использующего данный метод, как правило, значительно больше, чем для метода поиска кратчайших путей. Используйте данный метод, если вас интересует детальное исследование всех связей некоторого фрагмента ассоциативной сети.

Минимальная частота связи

Этот параметр задает минимальную частоту для каждой из связей в ассоциативной цепочке. К примеру, если задать минимальную частоту равную двум, то при поиске цепочка «Работа—1→ Успех—7→ Сла-

ва—1→ Подвиг» будет отброшена, потому что в ней содержится две связи с частотой 1. Тем не менее цепочка «Работа —20→ Труд—3→ Великий—2→ Подвиг» будет верной, потому что все связи которые в ней содержатся, имеют частоту более 2х.

Максимальная длина пути.

Этот параметр задает максимальную длину искомой ассоциативной цепочки. Значение длины пути сильно влияет на время обнаружения цепочек – чем больше его значение, тем длительнее время поиска.

Max длина пути	Только кратчайшие	Все пути
1-3	1-3 сек.	1-3 сек.
4	7 сек.	> 10 сек.
5-7	> 7 сек.	> 20 сек.

Искать пути только из двунаправленных связей.

При задании этого параметра активизируется особый режим поиска, при котором выводятся только те цепочки, которые состоят из двунаправленных связей. Если найденная цепочка содержит хотя бы одну связь, не являющуюся двунаправленной, то данная цепочка отбрасывается.

К примеру, цепочка *«Квартира ←1—1→ Новая ←1—1→ Улица»* будет отображена т.к. все связи в ней имеют как обратную, так и прямую направленность. Цепочка же *«Квартира—3→Моя —1→ Улица»* не будет отображена т.к. не имеет обратной цепочки *«Улица → Мой → Город»*.

Задание этого параметра позволяет выявить сильно связанные друг с другом слова. Следует использовать этот режим, если стоит задача избавиться от слабых связей в цепочках.

Запись связи имеет следующий формат:

стимул ←частота обратной связи — частота прямой связи→реакция

Выбор языка

В программе можно задать ассоциативный словарь, с которым вы работаете в данный момент. Для этого следует выбрать нужный язык из списка в левой части программы, на панели «Настройки». В текущей версии программы поддерживаются русский и английский языки. Изменение текущего языка работы программы затрагивает все функции программы: поиск ассоциативных цепочек, построение фрагмента сети, поиск реакций.

Построение фрагмента ассоциативной сети

Для работы с данной функцией программы следует переключиться на закладку «Фрагмент сети» в основном окне программы:

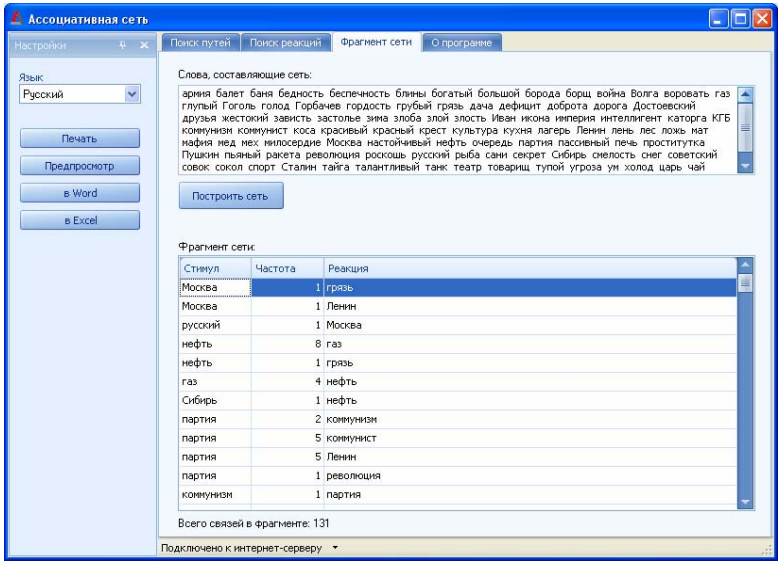


Рис.18. Поиск фрагмента АВС

Данный режим работы программы позволяет найти все ассоциативные связи для множества заданных стимулов, причем такие что и стимул и реакция будут каждой из найденных связей будут принадлежать множеству введенных слов. Таким образом вводится множество слов и программа находит все связи между ними, не учитывая те связи которые ведут к другим словам, не указанным во входном наборе.

Эта возможность позволяет выявлять сильно связанные фрагменты АВС.

Для того чтобы построить подсеть необходимо ввести несколько слов через пробел в поле «Слова, составляющие сеть» и нажать кнопку «Построить сеть». После этого отобразится результат построения фрагмента сети в виде множества ассоциативных связей.

Полученный результат можно распечатать или сохранить в формате Word, Excel, PDF или HTML.

Программа поддерживает построение фрагмента ассоциативной сети как для русского так и для английского языков.

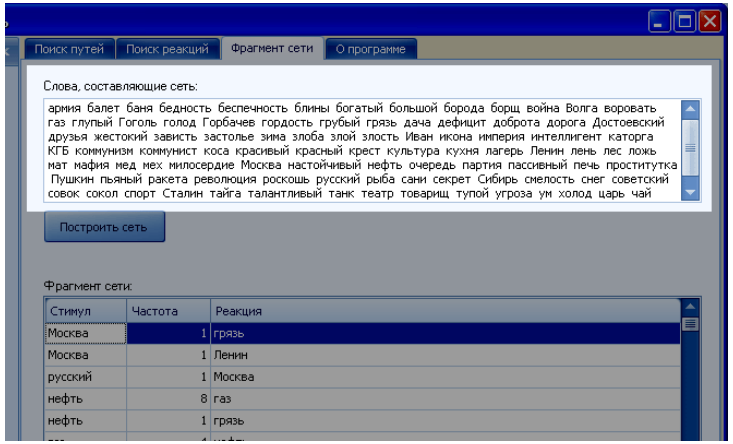


Рис.19. Ввод слов-стимулов

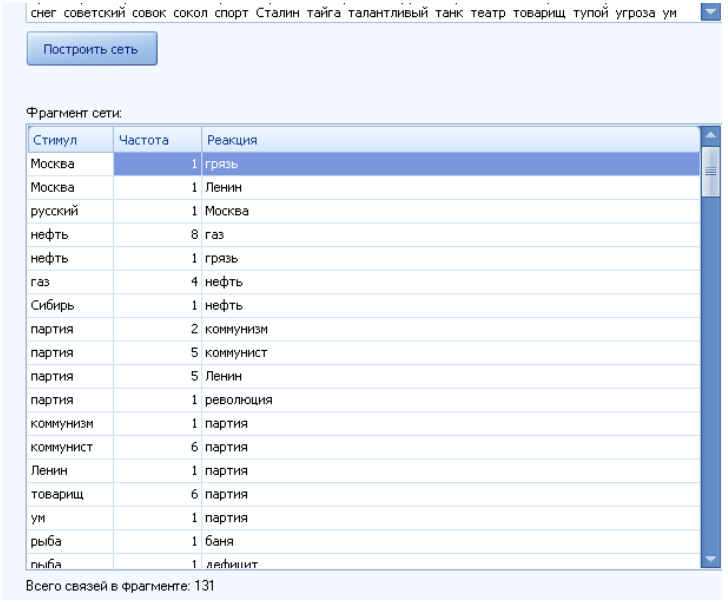


Рис.20. Результат построения подмножества ABC

Нахождение множества реакций для заданного стимула

Для работы с данной функцией программы следует переключиться на закладку «Поиск реакций» в основном окне программы:

Данный раздел программы позволяет найти все реакции, имеющиеся в словаре на заданное слово-стимул. Этот режим упрощает работу со словарем ассоциаций за счет того что поиск нужных словарных статей происходит более быстро.

Для нахождения всех реакций следует ввести в поле Стимул нужное слово и нажать кнопку «Искать». В таблице отобразиться перечень всех реакций на заданный стимул, с указанием частот.

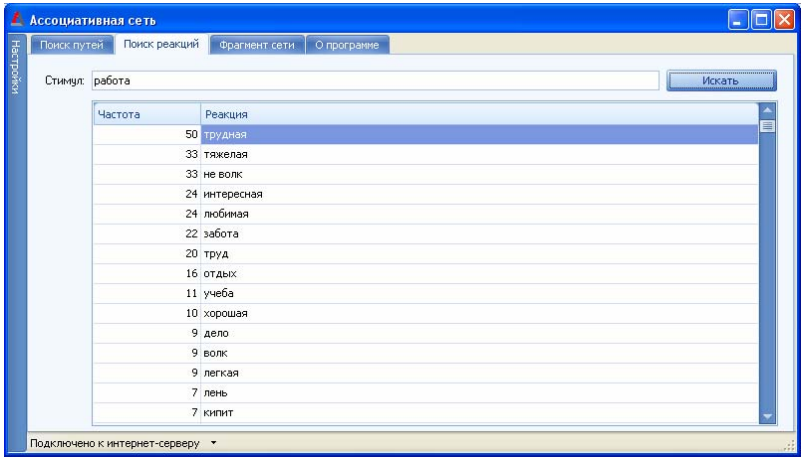


Рис.21. Поиск реакций на заданный стимул

Заключение

В результате была создана программа, которая может выполнять три основные функции: поиск ассоциативных цепочек в АВС, построение фрагмента АВС и поиск всех реакций на заданный стимул. Реализована возможность работы, как со словарем русского ассоциативного эксперимента, так и с английским.

Инструмент будет полезен профессиональным лингвистам, а так же людям творческих профессий, особенно работающих со словом. Кроме того программа «Ассоциативная Сеть» может быть использована в учебном процессе. К примеру, студенты МГТУ кафедры ИУ5 в рамках лабораторных работ по курсу «Семиотика Информационных Технологий» знакомились со структурой АВС с помощью данной программы.

Немаловажен тот факт что в ходе разработки программы был создан универсальный формат хранения данных для ассоциативных экспериментов, что позволит легко подключать новые ассоциативные словари к данной программе.

Литература

1. *Дейт, К., Дж.* Введение в системы баз данных, 7-е издание. Издательский дом «Вильямс», 2001.
2. *Ю.Н.Караулов.* Лингвистическое конструирование и тезаурус русского языка. Издательство «Наука», 1981.
3. *Кормен Т., Лейзерсон Ч., Ч. Риверс и др.* Алгоритмы: построение и анализ. Издательский дом Вильямс, 2007.
4. *Ананий В. Левитин.* Алгоритмы: введение в разработку и анализ. Издательский дом Вильямс, 2007.
5. *Дж. Люгер.* Искусственный Интеллект. Addison Wesley, 2005.
6. *А.Ахо, Дж.Хопкрофт, Д.Ульман.* Структуры данных и алгоритмы. Addison Wesley, 2005.
7. *Ю.Н.Караулов, Ю.Н.Филиппович.* Лингвокультурологический тезаурус русского языка. Москва, 2005.
8. *Ю.Н.Караулов.* Концептография языковой картины мира. Статья 1. Первый этап «восхождения» к образу мира: от элементарных фигур знания к предметно-референтным областям культуры. Сборник статей. «Азбуковник», 2004.
9. *Г.А.Черкасова*. Формальная модель ассоциативного исследования. Проблемы прикладной лингвистики. Выпуск 2. Сборник статей. «Азбуковник», 2004.
10. *Е.Е. Соколова.* Общая психология. Словарь. Москва, 2001.
11. *Kiss G.R., Armstrong C., Milroy R.* «The Associative Thesaurus of English». Edinburgh: Univ. of Edinb., 1972
12. *Ю.Н.Караулов, Г.А.Черкасова, Н.В.Уфимцева, Ю.А.Сорокин, Е.Ф.Тарасов* «Русский ассоциативный словарь. » в 2х томах, Москва, АСТ Астрель, 2002.
13. Дистрибутивный пакет Microsoft NET Framework 2.0 http://www.microsoft.com/downloads/details.aspx?displaylang=ru&FamilyID=0856EACB-4362-4B0D-8EDD-AAB15C5E04F5
14. Официальный сайт программы Ассоциативная Сеть http://www.associative.ru/

Д.В.Пономарев

МЕТОДЫ АНАЛИЗА СТРУКТУРЫ ТЕКСТА В ИСТОРИЧЕСКИХ РУКОПИСНЫХ ДОКУМЕНТАХ

Естественный язык долгое время был для лингвистов неприкосновенным объектом изучения, и работы в области языковедения носили описательных характер. Таким образом, язык рассматривался как некий самостоятельно развивающийся объект изучения. Рассматривались различные факты языка, проводилась систематизация, выделение его структур, схем функционирования. С развитием вычислительной техники, постановкой новых задач и накоплением знаний о языках появилась необходимость в лингвистических объектах, имеющих новые свойства. Позволяющих с другой стороны взглянуть на естественный язык. Не только на сложившиеся внешние признаки его функционирования, но и на принципы развития языка. Такими объектами могут быть порождающие синтезаторы, грамматики нового типа и т.д. [Караулов, 1981, с.19]

Распознавание текста является комплексной задачей, успешное решение которой зависит от многих факторов. В соответствии с предложенной группой NATO ASI [12] классификацией максимальной сложностью для распознавания обладают тексты, выполненные без пробелов между символами. При этом условии сложность распознавания слитного рукописного текста сравнима со сложностью текста выполненного печатными символами. Поскольку распознавание раздельных печатных символов согласно этой классификации сложности не представляет то ясно, что наиболее трудной задачей является разделение их между собой.

Традиционно задачу распознавания текста формулируют следующим образом. Множество объектов распознавания $\Omega=\{\omega_1, \dots \omega_z\}$ представляют собой входную последовательность фрагментов изображения, соответствующих отдельным символам текста. При этом в каждом фрагменте должно быть полное изображение только одного символа. Вводится в рассмотрение множество возможных вариантов разбиений данных объектов на классы $A=\{A_1, \dots A_r\}$ при этом считается, что если

выбран вариант разбиения A_α α=1, … ,r то множество Ω подразделяется на m_α классов эквивалентности

$$A_\alpha : \Omega_q^{A_\alpha} \cap \Omega_g^{A_\alpha} = \varnothing$$

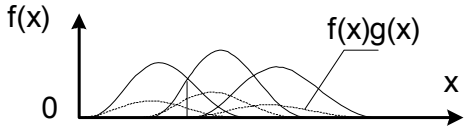
$$\cup_{i=1}^{m_\alpha} \Omega_i^{A_\alpha}, q, g = 1,...,m_\alpha, q \neq g$$

Если обучающая выборка достаточно представительна, то она позволяет определить условные плотности распределения значений признаков по классам

$$f_{\Omega_i}^{A_i}(X_1,...,X_N), i = 1,...,m$$

где $X_1,…,X_N$ – конкретные значения признаков фрагментов Ω={ω_1, … ω_z}. Функции $f_{\Omega_i}^{A_i}$ представляют собой плотность вероятности того, что при данном наборе значений признаков <$X_{1,…,}X_N$> фрагмент изображения представляет собой символ ω_1. То есть каждому возможному символу ставится в соответствие вероятность его появления при данных значениях измеренных признаков. Задача распознавания, таким образом, сводится к использованию Байесовского подхода для максимизации вероятности правильного распознавания фрагмента. Но в реальных условиях в измерении значения признаков тоже возможны ошибки. Это ошибки связанные как с погрешностью измерения признаков, так и с выделением самих объектов распознавания. Точность решения задачи распознавания, таким образом, ограничена степенью детализации определения характера распознаваемого объекта и точностью измерения признаков. Сама же точность определения характера объекта также является случайной величиной и ее влияние на условные плотности распределения признаков можно представить как $F_\Omega(X) = f_\Omega(x)g(x)$ где g(x)- не зависящая от Ω величина. Влияние g(x) на решение задачи распознавания выражается в снижении качества распознавания.

Таким образом, для успешной работы с текстом, необходимо уменьшить влияние этой величины, для чего нужен эффективный способ представления его к виду Ω={ω_1, … ω_z}.



Проверим эффективность использования традиционных методов анализа структуры текста, на примере одной из широко распространенных программ оптического распознавания текста Abbyy Fine Reader. В

этой программе реализован режим автоматического анализа структуры текста, который достаточно эффективно работает с текстом, выполненным типографским способом. Проверим, удастся ли этой программе определить наличие текста на одном из листов рукописи «Остромирово евангелие». Результат эксперимента представлен на рисунке 1 на котором ломаной линией выделены блоки текста и изображений.

Рис 1. Структура текста по результатам работы программы Fine Reader.

Можно заметить, что хотя алгоритм анализа структуры текста оперирует более крупными блоками текста, чем отдельные символы все же характер текста не позволил программе качественно их определить. Для работы с текстами древнерусских рукописей необходим более специализированный способ.

Попробуем определить требования к такому алгоритму, исходя из особенностей используемых в нем исходных данных. Качественный алгоритм определения структуры текста являющийся предварительным этапом в задаче распознавания, поэтому должен разделять символы и изображения на листе таким образом, чтобы обеспечить минимальное количество ошибок. При этом необходимо сохранить не только целостность фрагментов с изображением отдельных букв, но и их порядок. Каждый символ текста расположен на некоторой области изображения,

которую алгоритму необходимо выделить. Более того, символы в рукописи могут состоять из нескольких таких фрагментов, причем это может быть вызвано как особенностями начерка символа, так и крупными дефектами изображения, приводящими к нарушению их целостности. Необходимая конструкция может быть представлена в виде следующей списочной структуры.

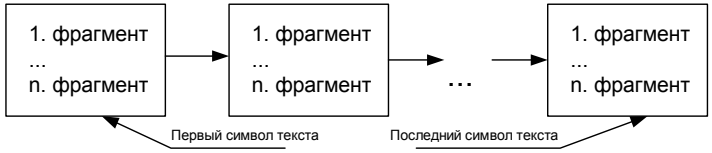


Рис 2 Необходимая структура текста

Для подобного представления данных необходимо правильно разделить символы как от декоративных и шумовых элементов, так и между собой. Порядок символов в строке обычно соответствует их положению на странице относительно левой границы листа но по оси Оу символ может занимать значительно пространство и даже перекрывать несколько последующих строк. Эта ситуация характерна для т.н. буквицы. Буквицей называют особым образом выделенную первую букву абзаца увеличенного размера. По высоте такая буква может превышать средний размер строки более чем в пять раз.

Чтобы преобразовать текст листа рукописи к удобному для машинного анализа виду, нужно в первую очередь разделить строки текста, учитывая эту особенность.

Одно из решений задачи разделения строк рукописного листа полууставной рукописи описано в [9]. В нем предлагается следующий алгоритм:

1). Определяется пороговое количество черных точек в линии растра, принадлежащей строке текста как среднее геометрическое максимального и минимального числа. Остальные строки растра считаются принадлежащими межстрочному пространству.

2). Лист делится на три части по горизонтали, в каждой из которых анализ проводится отдельно.

3). Из полученных таким образом строк удаляются те, которые по высоте больше удвоенной средней высоты строки и те, которые меньше некоторого фиксировано количества точек.

4).Удаляются те строки, которые расположены целиком только в одной из трех частей листа.

5). Полученные строки расширяются на некоторую долю от их высоты.

Можно заметить, что при построении этого алгоритма исходили из ряда допущений сильно снижающих точность анализа изображения. Например, предполагалось, что строка имеет постоянную ширину и отдельные строки можно разделить прямой линией. Несмотря на то, что буквы в полууставе действительно обычно отделены друг от друга, эту границу совсем не обязательно можно представить прямоугольником.

Основными следствиями из такой конструкции алгоритма анализа структуры текста является то, что вся информация о подстрочных и надстрочных знаках теряется, так как она находится в строках растра с недостаточным количеством черных точек. Кроме того, на шаге 4 теряется информация о строках, не превышающих по ширине 33% листа. На шаге 3 удаляются строки заголовка традиционно пишущиеся крупными буквами.

Чтобы избежать ошибок связанных с использованием такого рода упрощений при анализе рукописного листа следует ориентироваться на отдельные символы представленные некоторым количеством замкнутых контуров.

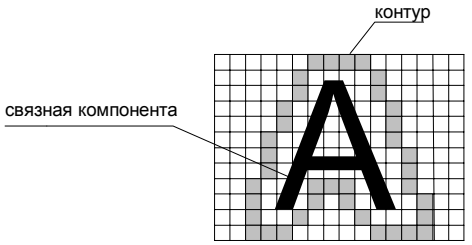


Рис 3. Контур символа

Замкнутый контур может быть определен как некоторое связное множество точек, ограничивающее область изображения, так что ни одна из этих точек не принадлежит области. Выделение замкнутых контуров ограничивающих связные компоненты листа можно провести с помощью алгоритма обнаружения границ Канни [10]. Такое представление изображения позволяет от аппроксимации строк прямоугольниками перейти к более точному их представлению как упорядоченной последовательности фрагментов изображения. Алгоритмы выделения границ чувствительны к шумовой составляющей изображения, поэтому перед применением этих алгоритмов необходимо очистить изображение

от мелких дефектов и привести изображение к виду удобному для анализа с помощью процедуры описанной в [5].

Замкнутый контур обладает рядом параметров, которые можно использовать для последующего изучения его структуры, поэтому представление структуры текста в виде множества замкнутых контуров, является более информативным, чем в виде множества точек. К этим параметрам относятся площадь и периметр контура.

Площадь замкнутого контура(S) - количество точек, которые им ограничены.

Периметр замкнутого контура(P) – количество точек принадлежащих границе области.

Некоторые параметры контура являются инвариантами относительно аффинных преобразований, поэтому их удобно использовать для анализа формы ограничиваемой контуром области. К таким параметрам относятся компактность и эксцентриситет.

$$c = \frac{P^2}{S};\ m_{ij} = \sum_{(x,y)\in Reg} (x-\bar{x})^i (y-\bar{y})^j;$$

$$e = \frac{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} - \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}.$$

Компактность – отношение квадрата периметра к площади.
Эксцентриситет(e) – мера нецентрированности контура.
У каждого контура есть центр определяемый как

$$\bar{x} = \frac{1}{n} \sum_{(x,y)\in Reg} x;\quad \bar{y} = \frac{1}{n} \sum_{(x,y)\in Reg} y;$$

Используя это соотношение, получаем точки центра всех контуров на листе.

Теперь необходимо восстановить порядок контуров и их целостность. Заметим, что точки расположены вдоль горизонтальных прямых соответствующих отдельным строкам и количество этих линий равно количеству строк на листе. Имея точное значение количества строк можно было бы выделить их с помощью алгоритма кластеризации К средних по оси Оу . Рассмотрим гистограмму распределения всех точек по оси ординат.

Из гистограммы видно, что количество строк можно определить по количеству локальных максимумов огибающей. Обозначим это число

Nc. Теперь используем полученную информацию для выделения отдельных строк.

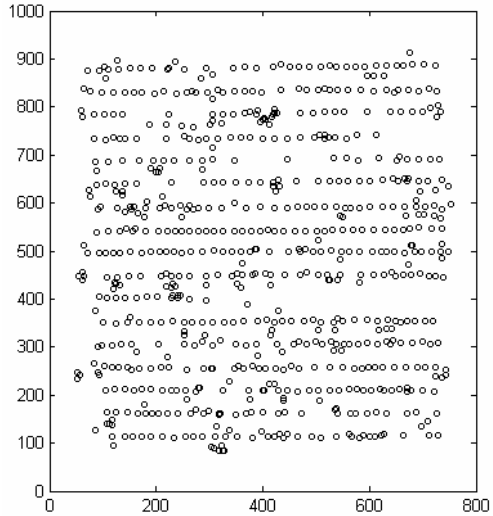


Рис 3. Координаты центра контуров листа

Пусть ординаты центров полученных контуров находятся в векторе X имеющей вид:

$$X=\begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

Определим характеристическую функцию с помощью матрицы разбиения:

U=[U_{ci}], u_{ci}={0,1}, c=1,Nc i=1,n Где c-й элемент вектора U указывает на принадлежность контура x_i к строке Ac.

Полученная матрица должна обладать свойствами

$$\sum_{i=1,k} u_{ci} = 1, c = 1, Nc$$

$$0 < \sum_{k=1,M} U_{ki} < Nc, i = 1..n$$

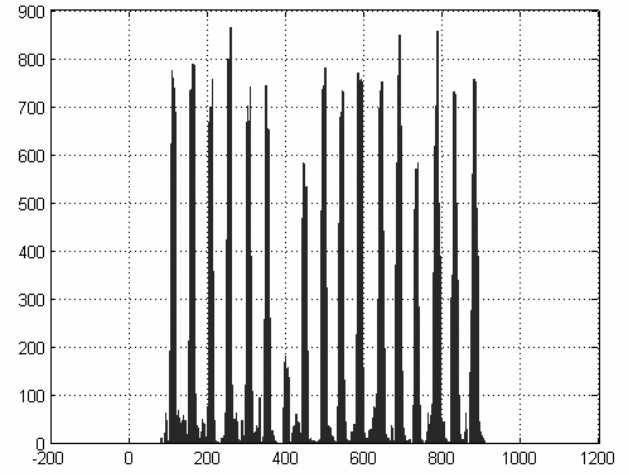


Рис 4. Распределение черных точек по оси Оу

Для оценки качества разбиения используется критерий

$\sum_{i=1,c}\sum_{Xc\in Si}\|V_i - X_c\|^2$, где X_c - ордината центра с-го контура;

Si={X_p}: u_{pi}=1,p=1,Nc – i-я строка.

$V_i=\frac{1}{|S_i|}\sum_{Xc\in Si} X_c$ - центр i-й строки.

Таким образом, задачу выделения строк можно сформулировать как следующую задачу оптимизации: найти матрицу U минимизирующую значение критерия

$$\sum_{i=1,c}\sum_{Xc\in Si}\|V_i - X_c\|^2$$

После проведения этой операции легко выделить строки по признаку количества сегментов в каждой из них. Например, из гистограммы на рис. 5. видно, что минимальный размер строки, который позволяет определить этот алгоритм, составляет четыре символа. Кроме того, каждому выделенному фрагменту соответствует его площадь, что позволяет учитывать только относительно крупные фрагменты символов, но не, например, сокращения методом суспенсии. Типичным элементом полууставного рукописного листа является знак титло, представляющий собой волнистую линию, ширина которой значительно больше высоты

применяемый для слов записанных с помощью метода сокращения - контракции. Подобные элементы легко обнаружить с помощью параметра эксцентриситета контура или его компактности. Минимальная компактность у круговых контуров, поэтому контуры, вытянутые по горизонтали или вертикали, отличаются большим значением этого параметра.

После проведения этапов предварительной обработки изображения и его сегментации необходимо задать разбиение A_α, определив множества классов эквивалентности Ω_i и поставив каждому из них в соответствие множество $S_i=\{s_1 \dots s_n\}$ связных областей изображения.

Структура текста задана следующими параметрами:

Nc – количество строк на листе рукописи

$M=(m_1 \dots m_N)$ – количество символов в каждой строке листа

$O:(i,j)=>\Omega$ – оператором ставящим в соответствие вектору координат символа рукописи класс эквивалентности представляющий элемент алфавита распознавания.

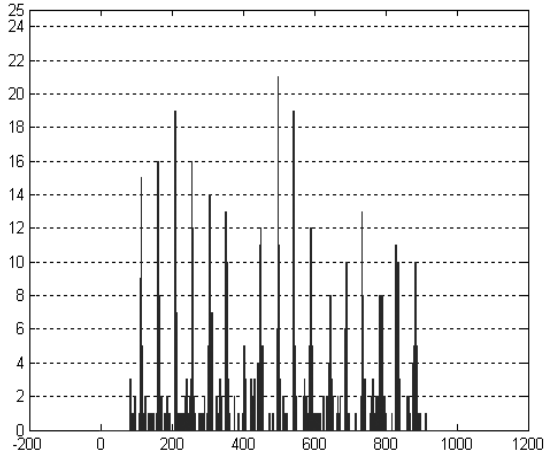


Рис. 5. Распределение контуров по координатам их центра.

Таким образом, для определения соответствия $S_i=>\Omega$ необходимо определить каждой паре <i,j> i:1..N, j:1..m_i множество S_i.

Шаг 1. – Разбить все множество контуров

$$S=\bigcup_{1..K} S_i; K = \sum_{i=1..N} m_i$$

на Nc подмножеств сегментов принадлежащих отдельным строкам. L_i i=1..N

Шаг 2. – Разбить каждое множество L_i на m_i подмножеств $S_{i,j}$ j=1..m_i соответствующее отдельным символам.

Шаг 3. – Определить каждой паре $<i,j>=>S_{i,j}=\{s_1 \ldots s_n\}$

***

В заключении можно отметить. Древнерусские рукописи обладают рядом особенностей, которые затрудняют их автоматизированный анализ. Необходимы специальные методы для выделения структуры рукописного текста, учитывающие эти особенности. Предложен алгоритм, позволяющий провести подобный анализ применительно к листам выполненным полууставным начерком.

Список литературы

1. *Айплатов Г.Н.* Русская палеография. 2003
2. *Горелик А.Л. Скрипкин В.А.* Методы распознавания. - Москва. Высшая школа 2004. 261 стр.
3. *Дьяконов В.* MATLAB 6.0 Обработка сигналов и изображений.
4. *Койнаш Б.В.* Регуляризующий байесовский подход в задачах классификации объектов по изображениям. Институт прикладной астрономии. Спб. 1991. 61 стр.
5. *Пономарев Д.В.* Предобработка изображений для системы распознавания исторических рукописных документов. // В кн.: “Интеллектуальные технологии и системы”. Вып 7.М.: МГТУ им. Баумана. С179-189.
6. *Розанов Ю.А.* Теория вероятностей, случайные процессы и математическая статистика. - 1989
7. *Славин О.А.* Комбинированные алгоритмы в задачах распознавания текстов. - Институт системного анализа РАН.
8. *Старков Ф.А.* Обработка изображений и распознавание образов. - Курск. 2003. 231 стр.
9. *Терещенко Л. А.* Вспомогательные исторические дисциплины: палеография. 2001
10. *Чикунов И. М.* Электронное издание древних рукописей и первопечатных книг.
11. *Шапиро Л.* Компьютерное зрение - Москва. Бином. 2006. 752 стр.
12. Shape in picture: mathematical description of shape in gray level images. - NATO ASI Berlin:Springer-Verlag 1994

А.А. Проскурнин

АВТОМАТИЗИРОВАННОЕ ИЗВЛЕЧЕНИЕ ДЕКЛАРАТИВНЫХ ЗНАНИЙ СУБЪЕКТА НА ОСНОВЕ КОГНИТИВНОГО ТЕЗАУРУСА

Введение

Статья посвящена проблеме извлечения теоретических знаний субъекта, относящихся к определенной предметной области в процессе взаимодействия субъект-компьютер: рассмотрена общая проблематика извлечения знаний субъекта, выделен определенный класс задач автоматизированного извлечения знаний субъекта, а также описаны основные идеи и концептуальные положения предлагаемого подхода к решению выделенного класса задач на основе применения «фигур знания».

Сразу необходимо уточнить, что задача извлечения знаний субъекта рассматривается здесь вне контекста известной в искусственном интеллекте и теории экспертных систем задачи приобретения знаний эксперта в определенной предметной области для их последующей формализации и хранения в базе знаний интеллектуальной системы. Приобретение (извлечение) и формализация знаний эксперта – это отдельная, специфическая задача, в первую очередь направленная на выявление «глубинных» знаний, которые применяются при решении той или иной конкретной задачи. Описанный в этой статье подход к извлечению знаний субъекта, напротив, в значительной степени ориентирован на работу с теоретическими, базовыми, декларативными знаниями определенной предметной области – это, в первую очередь, основные понятия предметной области и основные связи между этими понятиями. Такие знания составляют основу той или иной области знания и не связаны непосредственно с решением каких-либо конкретных задач.

Общая проблематика извлечения знаний субъекта

Перед тем как рассматривать подходы к решению задачи извлечения знаний субъекта, прежде всего нужно определить, в чем заключается природа сложности этой задачи, какие основные трудности препятствуют ее успешному решению.

Фундаментальная сложность проблемы извлечения знаний у человека состоит в том, что все ментальные, когнитивные процессы, мышление, память, работа сознания субъекта доступны для непосредственного восприятия только самому субъекту. Сознание, знания, память, мысли каждого человека всегда остаются для других людей «черным ящиком». Поэтому об уровне интеллектуального развития, содержании памяти, знаниях, способах мышления какого-либо человека можно судить только по каким-либо внешним проявлениям этих характеристик, которые для остальных людей выступают своеобразными «индикаторами» для формирования суждений. Естественно, это, прежде всего, деятельность человека, его речь, какие-либо предметы, созданные человеком и имеющие определенное смысловое наполнение. Здесь необходимо также учитывать, что механизмы работы сознания и памяти человека, процессы формирования его знаний являются в настоящее время во многом непознанными, парадоксальными и неподдающимися формализации. Например, в [1] отмечается «отсутствие хоть какого-нибудь приемлемого решения» проблемы сознания в психологии и рассматриваются очень многие парадоксы работы сознания и памяти человека, выявленные при проведении психологических экспериментов. В [2] приводятся самые различные аргументы для обоснования принципиально неалгоритмической природы работы человеческого сознания. В [3] утверждается, что «исчерпывающего описания структуры индивидуального знания не может быть дано принципиально».

Формирование суждений о знаниях другого человека имеет свою специфику по сравнению с формированием суждений о каких-то других его личностных качествах. В общем случае, при решении задачи извлечения знаний субъекта на основе взаимодействия субъект-компьютер можно выделить две группы проблем: 1) проблемы, возникающие независимо от того, решается задача извлечения знаний человеком или компьютером; 2) специфические проблемы, возникающие при извлечении знаний с помощью компьютера. Рассмотрим отдельно эти группы проблем.

Общие проблемы извлечения знаний субъекта

1. Проблема отсутствия информации, или проблема организации взаимодействия.

Чтобы сформировать суждения о знаниях субъекта, необходимо получить информацию об этих знаниях, иначе говоря, необходимо организовать такое взаимодействие с субъектом, в процессе которого субъект так или иначе проявит свои знания. Информация, полученная от субъек-

та, в которой выражаются его знания, может быть представлена в различной форме: числовой, текстовой, графической, звуковой. При формировании суждений о знаниях субъекта обязательно должна учитываться специфика используемой формы взаимодействия. В качестве возможных вариантов формы взаимодействия можно привести свободный диалог, анкетирование, интервьюирование, тестирование, игровую форму.

2. Проблема интерпретации полученной информации о знаниях субъекта.

Чтобы распознать, какие именно знания в той или иной предметной области присутствуют в информации, полученной от субъекта, необходимо, как минимум, самому обладать знаниями в этой предметной области. Помимо этого, разные субъекты (или компьютерные системы), обладающие знаниями о предметной области, могут сформировать различные суждения о знаниях субъекта в результате интерпретации полученной информации, в силу субъективности восприятия и разной «картины мира» в данной предметной области. Также, для того чтобы интерпретировать полученную информацию, нужно обладать знанием той знаковой системы, в которой представлена информация. Например, двое людей, обладающих знаниями в одной и той же предметной области, но говорящих на разных языках, не смогут обмениваться знаниями друг с другом.

3. Проблема достоверности формируемой модели знаний субъекта.

Чтобы оценить адекватность сформированных суждений о знаниях субъекта, нужно сравнить полученную модель его знаний с «реальными» знаниями. Так как «реальные» знания субъекта недоступны для непосредственного восприятия, такое сравнение провести невозможно. Каким образом в таком случае можно гарантировать достоверность, адекватность построенной одним субъектом (или компьютером) модели знаний другого субъекта? Одним из возможных методов обеспечения (повышения) достоверности является получение нескольких моделей знаний субъекта различными способами, например, на основе анализа различных видов информации, полученной от субъекта, или на основе различных форм взаимодействия. Тогда можно считать, что те знания субъекта, которые присутствуют сразу в нескольких моделях, имеют достаточно высокий уровень достоверности. Очевидно, что чем больше количество полученной от субъекта информации о том или ином элементе знания, тем более достоверным будет описание этого элемента в построенной модели знания. Также, достоверность зависит от того, на-

сколько хорошо знает предметную область и знаковую систему, в которой представлена информация, тот субъект (компьютерная система), который осуществляет извлечение знаний.

4. Проблема неполноты, неточности, нечеткости формируемой модели знаний.

Понятно, что построенная модель знаний субъекта всегда будет неполной в силу ограниченного времени взаимодействия, ограничений, накладываемых выбранной формой взаимодействия, массы случайных факторов, влияющих на процесс взаимодействия. На способность выражения знаний субъекта оказывают влияние его физиологическое и эмоциональное состояние, те или иные индивидуальные особенности субъекта: например, уровень знаний субъекта может быть высоким, но ясно и кратко выразить эти знания в вербальной форме ему (ей) трудно. При извлечении знаний человеком большую роль играет человеческий фактор, возможная предвзятость, субъективизм. Компьютер не может построить адекватную модель в силу примитивности интерфейса взаимодействия (по сравнению с взаимодействием людей), отсутствия у компьютера полноты знаний о предметной области, в силу эргономических ограничений. Поэтому при формировании модели знаний субъекта в той или иной степени нужно учитывать так называемые НЕ-факторы – неполноту, неточность, нечеткость, неопределенность и т.д.

Специфические проблемы извлечения знаний субъекта на основе взаимодействия субъект-компьютер.

1. Ограниченность интерфейса взаимодействия.

Человек способен воспринимать информацию с помощью различных органов чувств; для компьютера, на данный момент, основными устройствами ввода информации в процессе взаимодействия с субъектом, являются клавиатура и мышь.

2. Ограниченность знаний компьютера о той знаковой системе, в которой он получает информацию от субъекта.

Главную роль здесь играет ограниченность знаний компьютера о естественном языке. Обмен знаниями (опытом) между людьми и извлечение знаний одного человека другим происходят, прежде всего, при общении людей на естественном языке. Самый простой путь узнать о знаниях другого человека – попросить его выразить эти знания вербально, в письменной или устной форме. Поэтому применение методов понимания компьютером естественного языка – один из главных ключей к решению задачи извлечения знаний субъекта.

3. Ограниченность знаний компьютера о предметной области.

Понятно, что чем лучше компьютер знает предметную область, чем больше объем и качество его знаний, тем более эффективно можно организовать извлечение знаний субъекта в диалоге с компьютером. Эффективное представление знаний и их использование при решении задач, традиционно решаемых человеком, являются основным предметом исследования искусственного интеллекта. Безусловно, ограниченность знаний о предметной области присутствует и у человека, но в гораздо меньшей степени (здесь имеется в виду потенциальная возможность познания человека и компьютера).

Рассматриваемый класс задач автоматизированного извлечения знаний субъекта и возможные подходы к решению таких задач

Задача автоматизированного извлечения знаний субъекта может рассматриваться с самых разных точек зрения и иметь разные области практического применения. В данной статье рассматривается особый класс задач автоматизированного извлечения знаний субъекта, обладающий следующими свойствами:

1. Целью взаимодействия субъект-компьютер является извлечение базовых, теоретических, декларативных знаний субъекта в определенной предметной области. Такие базовые знания любой предметной области включают знание основных понятий этой предметной области и отношений между ними.
2. Субъект при взаимодействии с компьютером отвечает на относительно небольшое количество вопросов, связанных с определенным фрагментом (подобластью, темой) предметной области, причем ответ на эти вопросы субъект формирует на естественном языке.
3. Ответ субъекта на каждый вопрос представляет собой примерно одно предложение, то есть является достаточно кратким.
4. В результате взаимодействия с субъектом компьютер формирует модель знаний субъекта об определенном фрагменте предметной области, которая отражает знание основных понятий этого фрагмента предметной области и отношений между понятиями.

Основным отличительным свойством данного класса является формулирование ответов субъектом на естественном языке, таким образом, задача извлечения знаний субъекта в данном случае сводится к задаче извлечения знаний из текста на естественном языке, введенного субъектом, или, иначе говоря, к задаче преобразования текстовой информации в знания.

Обзор существующих подходов к пониманию естественного языка выходит за рамки настоящей статьи. Тем не менее, можно сразу отме-

тить, что многие из существующих подходов к анализу естественно-языковых текстов в различных ЕЯ-системах не подходят для решения данной задачи в силу различных причин. Например, частотные методы анализа семантики текста неприменимы в данном случае из-за слишком небольшого объема текста, который не позволяет анализировать различные частотные характеристики, такие как частота совместной встречаемости слов.

Также сразу можно утверждать, что решение данной задачи должно быть ориентировано на так называемые семантико-ориентированный и прагматико-ориентированный подходы к анализу естественного языка [4], иначе говоря, при анализе нужно максимально учитывать особенности выбранной формы взаимодействия, контекст задаваемого вопроса, особенности класса предметных областей, для извлечения знаний по которым будет применяться разработанное решение. Такое утверждение обосновывается тем, что создание универсальных средств анализа естественного языка и описания его семантики – крайне трудоемкая и, как показывает практика, вряд ли разрешимая задача. Реальное языковое разнообразие всегда оказывается шире искусственно созданных формализмов. Поэтому, для решения конкретной задачи целесообразно применять такие методы анализа ЕЯ, которые бы максимально учитывали специфику этой задачи.

Исходя из всего сказанного выше, на рис. 1. изображена схема взаимодействия основных компонентов (моделей и алгоритмов), необходимых для решения рассматриваемого класса задач. Главным компонентом является алгоритм интерпретации (преобразования), формирующий модель знаний субъекта исходя из полученной от него в процессе ответов на вопросы текстовой информации. Для решения этой задачи необходимы как знания о языке, так и знания о мире (предметной области).

Предлагаемый подход к решению рассматриваемого класса задач

Ниже кратко описаны основные идеи и концептуальные положения, которые составляют предлагаемый подход к решению рассматриваемого класса задач. Описание разбито на части, соответствующие различным компонентам схемы на рис. 1.

Знания о языке и предметной области

Основу предлагаемого подхода составляет применение когнитивного тезауруса, который представляет собой описание «интегрального», «усредненного» языкового сознания в данной предметной области в виде совокупности элементарных вербальных единиц знания – так на-

зываемых фигур знания. Такой подход к описанию языкового сознания, а также рабочие термины «когнайзер» и «фигура знания» предложены Карауловым Ю.Н. и описаны в его работах, например, в [5]. Сразу нужно указать, что в работах Караулова Ю.Н. речь идет об описании так называемого обыденного языкового сознания, наивной картины мира; идея описания когнитивного тезауруса в данной предметной области состоит в применении аналогичного подхода для описания языкового сознания в этой предметной области, являющейся частью научной картины мира.

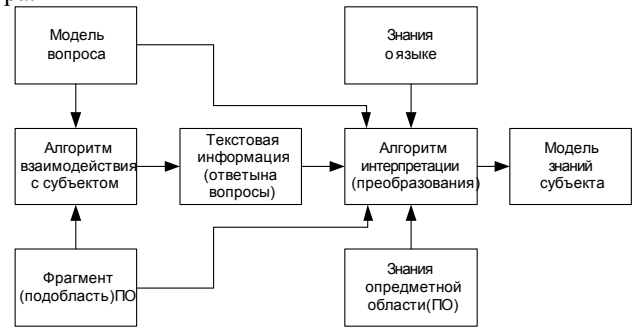


Рисунок 1. Схема взаимодействия основных компонентов, необходимых для извлечения знаний субъекта на основе анализа текста ответа

Фигура знания является пятикомпонентной структурой. Ее интенсионал, т.е. сущностное содержание, образуют три следующих компонента: знак – это слово или словосочетание, формула смысла – некоторое предложение, задающее смысл знака, и способ задания смысла, однозначно определяемый формулой смысла и знаком. Необходимо подчеркнуть, что способы задания смысла не зависят от знака, т.е. смысл одного и того же знака может быть задан различными способами. Например, смысл знак «информатика» может быть задан дескрипцией «дисциплина об информации», метафорой «дитя кибернетики», синонимом «computer science» или же, например, дефиницией «фундаментальная естественная наука, изучающая процессы передачи и обработки информации». Помимо указанных трех интенсиональных компонентов, в фигуру знания входят еще два компонента, определяющих ее экстенсионал – это когнитивная (референтная) область, к которой относится фигура знания, а также функция, определяющая ценность (вес) данной элементарной единицы знания. Когнитивная область представляет собой отражение в сознании субъекта той реальной предметной области, к

которой относится фигура знания; в случае обыденного языкового сознания совокупность таких когнитивных областей и формирует наивную языковую картину мира. При описании фигур знания когнитивного тезауруса в данной предметной области (ПО) научной картины мира, каждую фигуру знания можно отнести к той или иной подобласти (теме, разделу) внутри данной ПО. Например, внутри ПО «Информатика» можно выделить когнитивные области «Основные понятия», «Устройство компьютера», «Программирование», «Интернет» и т.д. Для характеристики ценности вербальной единицы знания Караулов Ю.Н. предлагает использовать два уровня: знание-рецепт, т.е. существенное, важное, основное знание, и знание-ретушь – второстепенное, фоновое, обладающее низкой ценностью знание. При этом фигура знания рассматривается не как некая статичная, застывшая структура, а именно в динамике, когда в разные моменты времени могут актуализироваться различные связи между рассматриваемыми пятью компонентами. Выделяется два основных режима работы когнайзера: активный и пассивный. В активном или смыслопорождающем режиме исходным является словесный знак; исходя из знака, когнайзер определяет некоторый смысл, и, тем самым, способ задания смысла. Цепочка параметров фигуры знания в активном режиме выглядит так: Знак → Способ → Смысл → Область → Функция. В пассивном режиме исходной информацией является формула смысла, в которой уже задан определенный способ задания смысла, а целью поиска когнайзера становится знак. Полную цепочку параметров фигуры знания для пассивного режима можно представить в следующем виде: Смысл → Способ → Знак → Область → Функция. В фигуре знания все пять компонентов связаны друг с другом, т.е. каждый компонент определенным образом зависит от четырех других. Схематически фигура знания изображена на рис. 2.

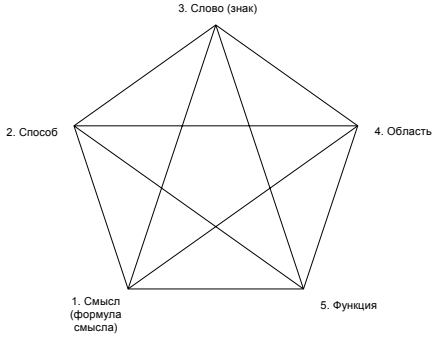


Рис. 2. Схематическое изображение фигуры знания [5].

Функционирование активного режима когнайзера описывается определенными траекториями движения в ассоциативно-вербальной сети. Взаимодействие параметров фигуры знания в активном режиме в работах Караулова Ю.Н иллюстрируется с помощью связей от стимула к реакции и от реакции к стимулу в ассоциативном тезаурусе русского языка [6]. В случае описания фигур знания в определенной предметной области научной картины мира, не основываясь на ассоциативных словарях (которых для большинства предметных областей не существует), для каждой фигуры знания можно, по крайней мере, выделить основные бинарные связи между знаком данной фигуры знания, и основными знаками-словами, выражающими понятия предметной области, и входящими в формулу смысла данной фигуры знания. Ниже в таблице даны некоторые примеры фигур знания в предметной области «Информатика» с выделенными для каждой фигуры знания связями.

Таблица 1. Некоторые примеры фигур знания в предметной области «Информатика»

Знак	Способ задания смысла	Формула смысла	Когнитивная область	Функция знания	Связи
Бит	Дескрипция	Единица измерения информации	Единицы измерения	Рецепт	Бит---Информация
Процессор	Метафора	Мозг компьютера	Устройство компьютера	Рецепт	Процессор---Компьютер
Информация	Дескрипция	Она передается от источника к приемнику в виде сигналов.	Основные понятия	Рецепт	Информация---Источник Информация---Приемник Информация---Сигнал
Бит	Дескрипция	Один разряд двоичного кода.	Единицы измерения	Рецепт	Бит---Двоичный код

Таким образом, когнитивный тезаурус, отражающий знания о языке и предметной области, можно представить в следующем виде: T = {E, R, F}. Здесь E – множество знаков (слов и словосочетаний), выражающих понятия данной ПО, R – множество отношений (связей) между понятиями ПО, F – множество фигур знания, актуализирующих те или иные отношения из R с помощью экспликации смыслов понятий из E через их связь с другими понятиями из E или понятиями смежных предметных областей и обыденной картины мира.

Модель вопроса

В качестве основных вариантов используемой формы вопроса можно предложить следующие формы:

1. Дайте толкование понятия E_i.

2. Кратко опишите, в чем состоит суть связи между понятиями E_1 и E_2.

3. Заданы два понятия: 1. E_1.2. E_2. Дайте толкование первого понятия с использованием второго понятия; это толкование должно отражать суть связи между понятиями.

Наиболее предпочтительным является использование третьей формы, так как она максимально сужает контекст, т.е. максимально четко определяет ожидаемый ответ в виде толкования смысла одного понятия с использованием другого понятия. Естественно, перед началом диалога (опроса) пользователю будут даны дополнительные пояснения в виде конкретного примера вопроса и примеров возможных ответов на этот вопрос. Множество фигур, знания, соответствующих (релевантных) такому вопросу, может быть разбито на 3 подмножества: 1) множество фигур знания, где знаком является E_1 или синонимы E_1; 2) множество фигур знания, где знаком является E_2 или синонимы E_2; 3) множество фигур знания, где знак отличается от знаков фигур знания предыдущих подмножеств, но при этом в формуле смысла используются как знак E_1 (или его синонимы), так и знак E_2 (или его синонимы). Здесь под синонимом понимается использование разных знаков для выражения одного и того же смысла, а не широкое понимание синонима как семантически близких слов.

Фрагмент (область) ПО

В соответствии с выбранной формой вопроса (третья форма) исходный фрагмент ПО представляется как семантическая сеть, а именно, в виде ориентированного графа, вершины которого соответствуют всем понятиям, которые будут использоваться в наборе вопросов, а дуги – самим вопросам. При этом дуга каждого вопроса направлена от соответствующего понятия E_1 вопроса к понятию E_2.

Алгоритм взаимодействия с субъектом

Алгоритм взаимодействия заключается в последовательном предъявлении всех вопросов, соответствующих дугам ориентированного графа, представляющего исходный фрагмент ПО, для которого проводится извлечение знаний. Порядок вопросов устанавливается экспертом до проведения контроля, таким образом, чтобы формулировка предыдущих вопросов исключала наличие подсказок для следующих вопросов.

Алгоритм интерпретации (преобразования)

Основная идея работы алгоритма интерпретации очень проста: для каждого вопроса выполняется процедура семантического сравнения текста ответа на вопрос с каждой из фигур знания, входящих в множество фигур знания, соответствующих (релевантных) вопросу. Для данного ответа и данной фигуры знания (ФЗ) результат выполнения процедуры семантического сравнения состоит в подтверждении (опровержении) следующих гипотез: 1) Какое-либо совпадение ответа и ФЗ отсутствует; 2) Ответ включает ФЗ; 3) ФЗ включает ответ; 4) Ответ и ФЗ эквивалентны; 5) Ответ и ФЗ частично пересекаются. При выполнении процедуры могут учитываться параметры фигуры знания, например, способ задания смысла, а также определенные знания о вопросе. Например, в зависимости от типа связи между понятиями в вопросе для каждого из вопросов может быть описан соответствующий фрейм. Так, связи «информатика - информация» соответствует фрейм «Описание науки», а связи «бит - информация» соответствует фрейм «Описание единицы измерения». В целом при работе процедуры семантического сравнения могут быть в той или иной степени реализованы морфологический и синтаксический анализы. Разработка конкретного алгоритма, реализующего эту процедуру, является предметом дальнейших исследований.

Модель знаний субъекта

В соответствии с описанной выше идеей алгоритма интерпретации модель знаний субъекта может быть представлена в двух различных «проекциях». Во-первых, это множество фигур знания, для каждой из которых в той или иной степени подтверждено присутствие данной фигуры в языковом сознании субъекта. Например, для каждой из фигур знания, входящих в модель знаний субъекта, может быть вычислен некий фактор уверенности в том, что данная когнитивная единица входит в знание субъекта. Во-вторых, учитывая параметры фигуры знания (прежде всего знак) и связи между понятиями ПО, соответствующие каждой из фигур знания, на основании первой «проекции» может быть синтезирована другая, представляющая собой семантическую сеть, от-

ражающую те понятия и отношения, знание которых в той или иной степени было продемонстрировано субъектом в процессе взаимодействия.

Заключение

В статье представлены только основные идеи и концептуальные положения предлагаемого подхода к решению выделенного класса задач. Основное преимущество данного подхода состоит в том, что в подавляющем большинстве аналогичных систем, организующих анализ ЕЯ для извлечения «базовых» знаний, модель знаний субъекта получается за счет сравнения либо ЕЯ-формулировок ответа, либо их концептуального представления с некоторой «эталонной», «правильной» моделью знаний. При сравнении данной формулировки с эталонными учитывается только достаточно ограниченное число формулировок, данных экспертами. При преобразовании ЕЯ-текста к концептуальному представлению крайне трудно учесть такие способы задания смысла, как метафора, цитата, суждение, ассоциация и др. Предлагаемый подход основан на построении не «эталонной», а реальной, «живой» модели языкового сознания в данной предметной области, отражающей самые различные возможные способы задания смыслов понятий разными субъектами, начиная от ассоциации и метафоры, и заканчивая строгой дефиницией. В такую модель языкового сознания одновременно входят и «книжное», академическое знание, и знание, выражаемое на «обычном», собственном языке субъекта. Для реализации предлагаемого подхода необходимо вести работу по следующим основным направлениям: 1) накопление фигур знания в определенной предметной области; 2) накопление реальных ответов субъекта на выбранные формы вопросов; 3) исследование применимости к данному подходу различных математических моделей и лингвистических процедур анализа текста. Одной из наиболее очевидных областей применения данного подхода является сфера образования.

Литература

1. *Аллахвердов В.М.* Методологическое путешествие по океану бессознательного к таинственному острову сознания. – СПб.: Издательство «Речь», 2003. – 368 с.

2. *Пенроуз Р.* Новый ум короля: О компьютерах, мышлении и законах физики: Пер. с англ. / Общ. ред. В.О. Малышенко. Предисл. Г.Г. Малинецкого. Изд. 2-е, испр. – М.: Едиториал УРСС, 2005. – 400 с. (Синергетика: от прошлого к будущему).

3. *Александров И.О., Максимова Н.Е.* Закономерности формирования нового компонента структуры индивидуального знания. // Психологический журнал, 2003, том 24, № 6, с. 55-76.

4. *Сулейманов Д.Ш.* "Аналитический обзор отечественных и зарубежных работ в области обработки естественного языка в аспекте прагматически-ориентированного подхода". // Электронный журнал Казанского госуниверситета "Информационные технологии". - Казань, 1999.

5. *Караулов Ю.Н.* О единицах знания. // Полифония образования и англистика в мультикультурном мире. Тезисы первой международной конференции Ассоциации англоведов и преподавателей английского языка 25-26 ноября 2003 г. Москва, МГЛУ, 2003.

6. *Караулов Ю.Н., Сорокин Ю.С., Тарасов Е.Ф., Уфимцева Н.В., Черкасова Г.А.* Русский ассоциативный словарь. Т. 1-6, М., 1994-1998.

7. *Филиппович Ю.Н., Черкасова Г.А., Дельфт Д.* Ассоциации информационных технологий: эксперимент на русском и французском языках. С предисловием Уфимцевой Н.В. М.: МГУП, 2001. 304 с. – Книга в комплекте с CD-ROM.

8. *Филиппович Ю.Н., Прохоров А.В.* Семантика информационных технологий: Опыты словарно-тезаурусного описания. С предисловием А.И. Новикова М.: МГУП, 2002. 368 с. – Книга в комплекте с CD ROM.

9. *Шаров Д.А.* Система анализа формулировок. // Статьи, принятые к публикации на сайте международной конференции Диалог' 2004.

10. *Гаврилова Т.А., Червинская К.Р.* Извлечение и структурирование знаний для экспертных систем. – М.: Радио и связь, 1992.

11. *Попов Э.В.* Общение с ЭВМ на естественном языке. Изд. 2-е, стереотипное. – М.: Едиториал УРСС, 2004. – 360 с. (Науки об искусственном).

12. *Черепанова Ю.Ю.* Тестирование теоретических знаний на основе применения тезауруса семантических полей для построения концептуальных моделей текста ответа обучаемого на естественном языке. // Материалы международной научно-методической конференции «Образование и виртуальность». - Ялта: ЯИМ, 2001.

13. *Всеволодский С.Н., Гаврилов А.В.* Архитектура интеллектуальной системы тестирования знаний с анализом ответов на естественном языке. // Международная конференция ИСТ-2003 "Информационные системы и технологии", Новосибирск, 2003. - Т.3, С. 114-115.

А.С. Сигачёв

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА МОДЕЛЕЙ ПРЕДСТАВЛЕНИЯ ТЕКСТОВ

В данной статье проводится экспериментальное сравнение различных моделей представления текста применительно к задаче анализа тематической близости текстов небольшой коллекции. Исследуются параметры: вид формулы TF-IDF, метод отбора слов для базиса векторного пространства, меры расстояния между текстами в коллекции. В качестве критерия качества используется отношение меры близости текста словарной статьи к аналогичной словарной статье из другого словаря.

Введение

Одним из основных методов информационного поиска является представление документа в виде числовой векторной модели (vector space model), в которой документы рассматриваются как неупорядоченные наборы слов, при этом не учитывается информация о положении слов в тексте и используя только их частоты.[1] При построении подобных моделей, часто применяется метод TF-IDF, являющийся статистической мерой, используемой для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса. Его суть заключается в том, что вес некоторого слова пропорционален количеству употребления этого слова в документе, и обратно пропорционален частоте употребления слова в других документах коллекции. С момента предложения формулы TFIDF в середине 1970-х годов, предлагаются различные модификации этой формулы, одной из самых известных является BM25 [2], имеющая несколько коэффициетов определяемых разными исследователями самостоятельно. В данной работе сравнивается эффективность различных вариантов формулы TF-IDF применительно к задаче кластеризации документов небольшой текстовой коллекции, размером до 200 документов. Помимо выбора формулы для оценки веса слова в документе, исследуются другие этапы построени модели, оказывающее влияние на машинное обучение: методы сокращения пространства признаков, выбор беры близости векторов в векторном пространстве.

Основным допущением эксперимента являлось то предположение, что метод построения числовой модели текстов и определения меры близости векторов, который близко расположит в выходном пространстве признаков вектора текстов, про которые заведомо известно, что они тематически близки, будет наилучшим образом подходить для проведения кластеризации других текстов коллекции.

Задача и описание эксперимента

Для проведения эксперимента были отобраны 100 энциклопедических статей по тематике "Физика" размером от 500 до 6500 слов: 50 статей были взяты из Большой Советской энциклопедии издания 1969—1978 годов, 50 статей из русской версии популярной сетевой энциклопедии «Википедия», из них 30 статьям каждой энциклопедии соответствовали 30 статей другой энциклопедии, остальные статьи не имели прямых аналогов.

Табл.1 Тематика и размер статей

Википедия	**Большая Советская энциклопедия** (3 изд.)
Акустика (502 слов)	Акустика (2030 слов)
Общая теория относительности (6511 слов)	Относительности теория (6326 слов)
Уравнение Максвела (1399 слов)	Лоренца-Максвела уравнение (644 слов)
Гамильтонова механика (801 слов)	
	Галилея принцип относительности (529 слов)

Каждая статья была проиндексирована локальной поисковой системой Lucene. В дальнейшем для определения основных статистических характеристик текста использовались стандартные функции ПО Lucene для работы с индексами. Это позволило избежать самостоятельной реализации механизмов определения формата документа, кодировки текста, выделения слов в документе, приведения слов к нормальной форме, подсчёта количества слов и документов, в которых они встречаются. Стоит заметить, что поисковая система Lucene не производит морфоло-

гический разбор слов для приведения к нормальной форме, а использует более простой механизм «стеммера» — отбрасывания известных алгоритму окончаний и суффиксов для получения основы слова. В эксперименте анализировались только слова документов имеющие не менее 3 символов и состоящие только из букв русского алфавита.

Для каждого слова каждого документа был рассчитан вес по 9 вариантам формулы TF-IDF, 4 способами были рассчитаны меры близости между документами коллекции, для построения базиса векторного пространства отбиралось различное количество слов — 9 вариантов, четыремя разными методами. Таким образом, в сравнении участвовали 1296 моделей текстовый коллекции.

Варианты TF-IDF

В работе оцениваются следующие методы расчёта веса слов.

1. Формула Салтона

Одним из первых, кто предложил использовать метод TF-IDF был Герард Салтон. В своей работе [3] он предлагает следующую формулу, которая в дальнейшем использовалась многими исследователями.

$$W_{ij} = TF_{ij} \cdot \log\left(\frac{D}{DF_i}\right)$$

Здесь TF_{ij} — частота терма i в документе j, D — число документов, DF_i — число документов, в которых встречается терм i.

2. Логарифмическая версия TF-IDF

Некоторые исследования [4] показывают, что использование логарифмических значений частоты терма положительно сказывается на результатах кластеризации текстов. Для оценки предлагается формула:

$$W_{ij} = \log\left(0{,}5 + TF_{ij}\right) \cdot \log\left(\frac{D}{DF_i}\right)$$

3. Нормализация частоты терма

Некоторые участники семинара РОМИП [5] в качестве эталонной формулы TFIDF используют формулу с нормализацией частоты термов, описанную в учебнике Эда Гринграсса [6]

$$W_{ij} = \left(0{,}5 + \frac{TF_{ij}}{2 \cdot MTF_j}\right) \cdot \log\left(\frac{D}{DF_i}\right)$$

где MTF_j — максимальная частота среди всех термов документа j.

Другой способ нормализации , описанный в том же учебнике

$$W_{ij} = \frac{1 + \log(TF_{ij})}{1 + \log(ATF_{ij})} \cdot \log\left(\frac{D}{DF_i}\right)$$

где ATF_j — средняя частота термов в документе j.

4. Расчёт TF-IDF по методу поисковой системы Lucene

Одной из важных функций поискового программного обеспечения является ранжирование результатов поиска по релевантности запросу пользователя. Одним из этапов алгоритма, реализующего данное ранжирование является определения веса слова в документе. Так как поисковый сервер Lucene распространяется свободно в исходных текстах, есть возможность изучить и оценить используемый им алгоритм. [7]

$$W_{ij} = \sqrt{TF_{ij}} \cdot \left(1 + \log\left(\frac{D}{1 + DF_i}\right)\right)^2 \cdot N_j$$

где N_j — фактор, нормирующий дляну документа, так, у документов с малым числом термов он будет выше, а у документов с большим числом — ниже.

$$N_j = \frac{3}{\log(1000 + NT_j) \cdot \log(10)}$$

где NT_j — количество термов в документе j.

5. Формула Яндекса на основе Bm25

На семинаре РОМИП-2006 Илья Сегалович представил исследование [8], в котором рассматривалось несколько методов определения веса терма. Первая формула имеет следующий вид

$$W_{ij} = \log\left(\frac{TT}{CF_i}\right) \cdot \frac{TF_{ij}}{TF_{ij} + 1 + \frac{1}{350 \cdot NT_j}}$$

где TT — общее число термов в коллекции, CF_i — количество терма i во всей коллекции.

6. Формула Яндекса на основе P_OnTopic

Другой представленной Яндексом формулой была

$$W_{ij} = \log\left(1 - e^{\frac{-1{,}5 \cdot CF_i}{D}}\right) \cdot \frac{TF_{ij}}{TF_{ij} + 1 + \frac{1}{350 \cdot NT_j}}$$

7. Простые методы

Для сравнение в исследование также были включены простые методы оценки веса слов. Бинарный метод показывает есть ли указанный терм в документе.

$$W_{ij} = 1, t_i \in D_j$$

$$W_{ij} = 0, t_i \notin D_j$$

Частота терма в документе

$$W_{ij} = TF_{ij}$$

Варианты дистанции

Помимо выбора формулы для определения терма слова в в векторе, характеризующим документ, важную роль играет выбор меры определения близости между векторами в модели векторного пространства.

1. Евклидово расстояние между векторами

$$L = \sqrt{\sum_{i=1}^{N} (w_{ai} - w_{bi})^2}$$

2. Косинусная мера

Ищется косинус угла между векторами. В отличие от евклидова расстояния данная мера не учитывает размер векторов документов.

$$L = \frac{\sum_{i=1}^{N} (w_{ai} \cdot w_{bi})}{\sqrt{\sum_{i=1}^{N} (w_{ai})^2} \cdot \sqrt{\sum_{i=1}^{N} (w_{bi})^2}}$$

3. Расстояние Чебышева

Примером пространства с данной метрикой служит поле с квадратными ячейками, на котором фишки (например шахматный король) могут двигаться в любом направлении, в том числе по диагонали.

$$L = max\left(\left|w_{ai} - w_{bi}\right|\right)$$

4. Манхеттеновское расстояние

Данный метод получил название из задачи о маршруте такси в городе, в котором все улицы пересекаются под прямым углом. При необходимости движения «по диагонали» маршрут прокладывается «ступеньками».

$$L = \sum\left(\left|w_{ai} - w_{bi}\right|\right)$$

Выбор признаков

Важным этапом построения векторной модели корпуса текстов является сокращение размерности пространства признаков. Из всех характеризующих документ слов и их весов следует отобрать только наиболее характерные. Это способствует увеличению производительности при проведении дальнейшей обработки, а кроме того, при правильном выборе критериев отбора признаков, позволяет увеличить точность итогового результата за счёт отбрасывания общеупотребительных слов «зашумляющих» текст.

В данной работе было рассмотрено три метода уменьшения размерности пространства признаков:

- В качестве компонентов векторного пространства выбирались термы, вес суммарный вес которых во всех документах максимальный.
- Выбирались термы с максимальным суммарным весом, но отбрасывались 20 или 100 первых значений в списке. Данный метод основывается на предположении, что термы получившие наибольший вес являются общеупотребительными и не должны оказывать положительного влияния на сравнение текстов.
- Из всего набора термов выбирались термы с наибольшей дисперсией веса. Можно предположить, что подобные термы в наибольшей степени описывают различие документов коллекции.

Для каждого метода были построены модели, имеющие размерность 20, 50, 100, 200, 400, 600, 800, 1000, 2000 термов.

Критерии оценки

Основным критерием сравнения описанных выше вариантов определения меры близости текстов было выбрано число ошибочно определённых наиболее близких по смыслу текстов. Так, если в результате работы алгоритма статье Большой Советской энциклопедии оказыва-

лась наиболее близка статья Википедии, определямая из таблицы соответствия статей, но засчитывалось правильное срабатывание алгоритма, иначе — засчитывалась ошибка.

Помимо основного критерия сравнения использовались дополнительные критерии, отсеивающие варианты алгоритма, работающие недостаточно точно. Так, не рассматривались варианты построения моделей, которые:

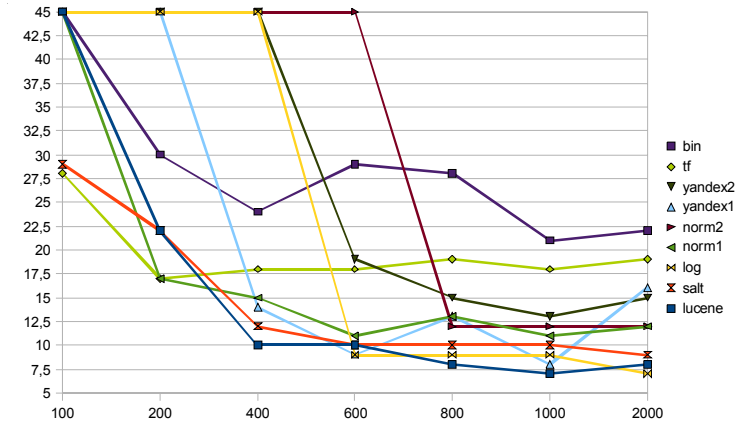
представляли два разных документа в виде одинаковых векторов;

представляли статьи на одну тематику ортогональными векторами.

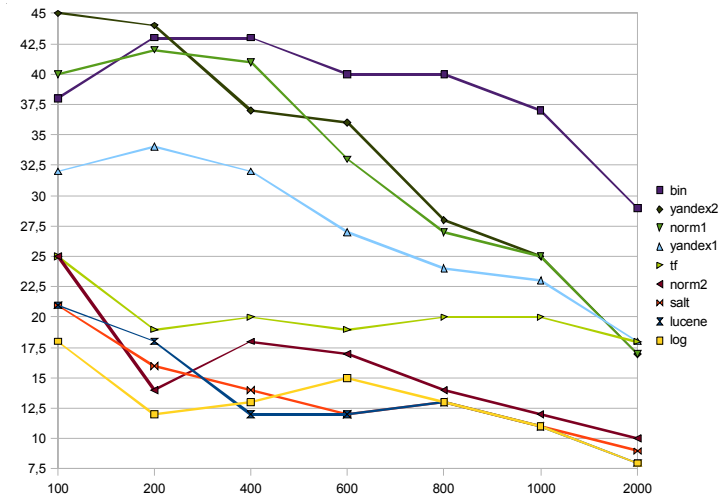
В качестве дополнительного критерия рассматривалось отношение расстояния между статьями соответствующей тематики и средним расстоянием до других статей.

Результаты экспериментов

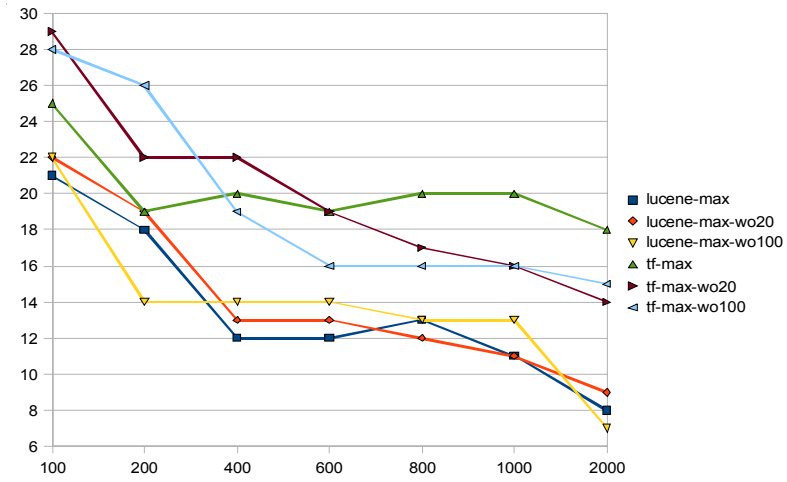
Эксперимент показал, что наименьшее количество ошибок при работе алгоритмов сравнения текстов происходит при использовании косинусной меры, поэтому остальные меры определения расстояния подробно не анализировались.



Сравнение алгоритмов определения веса при выборе термов с максимальной суммой весов



Сравнение алгоритмов определения веса при выборе термов с максимальной дисперсией



Сравнение алгоритмов определения при различных вариантах фильтрации термов с максимальным весом

Выводы

- После сравнения результатов различных методов сравнения тематической близости текстов небольшой текстовой коллекции, можно сделать выводы:
- Векторы размерностью меньше 100 не соответствуют установленным критериям качества.
- При сравнении векторов косинусную меру можно считать безальтернативной.
- Наилучшие результаты продемонстрировали методы log (Логарифмическая версия TF-IDF), lucene (алгоритм поискового движка Lucene) и salt (оригинальна).
- Метод отбора признаков по максимальной дисперсии даёт приемлемые результаты только начиная со значений 400 термов
- Отбрасывание при построении базиса векторного пространства термов имеющих максимальный суммарный вес является оправданным только для простых алгоритмов вычисления веса.
- Использование морфологического анализатора от Яндекса вместо стеммера качественно не влияет на результаты, количество ошибок снижается на 5-7%.

Литература

1. Сигачёв А. С. Модель текста в виде набора числовых признаков //Интеллектуальные технологии и системы. №7 — М., 2006.
2. S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In TREC-3, 1994
3. G. Salton, C. Buckley, Term Weighting Approaches in Automatic Text Retrieval, Information Processing and Management, Vol. 24, No.5, P513,
4. Ciya Liao, Shamim Alpha, Paul Dixon. Feature Preparation in Text Categorization
5. Кондратьев Михаил Е. Двухуровневая иерархическая кластеризация новостного потока в РОМИП 2006
6. E. Greengrass, Information retrieval: A survey. DOD Technical Report TR-R52-008-001, 2001
7. Apache Lucene-Scoring // http://lucene.apache.org/java/docs/scoring.html
8. Андрей Гулин, Михаил Маслов, Илья Сегалович. Алгоритм текстового ранжирования Яндекса на РОМИП-2006
9. Yang: A Comparative Study on Feature Selection in Text Categorization. // Тез. докл. ICML-97, 14th International Conference on Machine Learning — Nashville, 1997

Авторефераты квалификационных работ

А.А.Александрова

МУЛЬТИМЕДИЙНЫЙ ТЕЗАУРУС ЖЕСТОМИМИЧЕСКИХ И АРТИКУЛЯЦИОННЫХ ОБРАЗОВ ТЕРМИНОВ ИНФОРМАТИКИ[7]

Общая характеристика работы

Актуальность темы. В ходе профессиональной деятельности людей с нарушением слуха, связанной с областью «Информатика» (в частности, в процессе преподавания дисциплин, связанных с областью «Информатика»), возникают проблемы, препятствующие эффективному обмену информацией как в процессе межличностной коммуникации глухих и плохослышащих, так и в процессе их общения со слышащими людьми. Ниже перечислены факторы, обуславливающие возникновение данных проблем.

1. Отсутствие единства жестового обозначения терминов области «Информатика»: так, в процессе обучения студентов с нарушением слуха в МГТУ им. Н.Э. Баумана, ряд терминов информатики означивается либо самими студентами, либо сурдопереводчиком, причем полученные обозначения, как правило, используются в

---

7 Статья представляет собой материалы автореферата диссертации на соискание степени магистра техники и технологий, выполненной на кафедре «Системы обработки информации и управления» Московского государственного технического университета им. Н.Э. Баумана в 2007 г. Руководитель диссертации к.т.н., доц. Ю.Н.Филиппович; консультант к.т.н., зам. зав. каф. ИБМ-6 МГТУ им. Н.Э. Баумана В.А.Дадонов; рецензенты: зав. сектором механики и управления движением робототехнических систем отдела динамики космического полета ИПМ им. М.В. Келдыша РАН, лауреат Ленинской и Государственных премий СССР проф., д.ф.-м.н. А.К. Платонов; с.н.с., к.ф.-м.н. ИПМ им.М.В.Келдыша РАН; доцент каф. ФН-12 МГТУ им.Н.Э.Баумана А.А. Кирильченко.

рамках одной спецгруппы. Следствием этого является возникновение ряда синонимичных жестовых обозначений термина, а также ряда жестовых омонимов.

2. Вторая проблема обусловлена отсутствием жестовых эквивалентов у большинства терминов информатики (к таким терминам относятся, например, названия программых продуктов, языков программирования и т.д.) Данные термины воспроизводятся при помощи знаков дактильного алфавита. Такое представление затрудняет восприятие и уменьшает скорость передачи информации, например, в условиях лекционных работ.
3. Еще одна проблема связана с аналитическим способом формирования ряда жестов, заимствованных из РЖЯ. «В то же время в лексическом составе РЖЯ много жестов, передающих значения аналитически, расчленено. При помощи такого рода обозначений передаются смыслы ‘мебель’: СТОЛ СТУЛ КРОВАТЬ РАЗНЫЙ: ‘овощи’: КАРТОФЕЛЬ КАПУСТА ОГУРЕЦ РАЗНЫЙ и др. Расчлененность ярко выражена в условиях, когда требуется выразить смысл, для которого нет готового жеста. Например, для наименования ягоды черники используется жестовая конструкция: ЯГОДА ЕСТЬ ЯЗЫК ЧЕРНЫЙ; для значения ‘бирюзовый’ — например СИНИЙ ОТРИЦАНИЕ (ЗЕЛЕНЫЙ ОТРИЦАНИЕ) СМЕШАТЬ» [5]. Такое представление уменьшает скорость передачи информации, ведет к неоднозначности жестового обозначения терминов информатики.
4. По мнению ряда преподавателей и сурдопедагогов [3] большинство глухих и плохослышащих студентов, в силу специфики своего мышления, обладают небольшим словарным запасом, причем значения многих слов им либо не знакомы, либо понимаются посвоему. Таким образом, в ходе лекционных работ, приходится пояснять значения не только технических терминов, но и ряда других слов, приводя большое количество примеров. Кроме того, общий уровень подготовки у глухих и плохослышащих студентов ниже, чем у студентов с нормальным слухом.

Перечисленные выше проблемы приводят к идее создания некоего пополняемого мультимедийного «хранилища» уже существующих жестовых обозначений терминов информатики, в котором в явном виде указаны семантические отношения между составляющими его единицами. Таким «хранилищем», в соответствии с [6], является тезаурус.

Проблема отсутствия единства жестового обозначения терминов информатики решается возможностью добавления в тезаурус сущест-

вующих диалектический жестовых обозначений терминов информатики.

Проблема отсутствия жестового обозначения ряда терминов информатики решается путем заимствования для обозначения данных терминов уже существующих в КЖР или РЖЯ жестовых обозначений или путем конструирования нового жеста.

Главное же структурное свойство тезауруса – наличие более чем одного входа, частично решает проблему отсутствия единого понимания значений терминов информатики.

Объект исследования. Объектом исследования являются жестомимические обозначения терминов информатики.

Предмет исследования. Предметом исследования являются принципы проектирования жестомимического тезауруса.

Цели исследования. Целями исследования являются:

1. анализ структуры жеста;
2. разработка варианта жестовой нотации;
3. разработка подходов к построению жестомимического тезауруса;
4. проектирование структуры мультимедийного тезауруса жестомимических и артикуляционных образов терминов информатики;
5. разработка макета системы.

Апробация работы. Содержание отдельных разделов и диссертации в целом было доложено на студенческой научной конференции, проводимой в МГТУ им. Н.Э. Баумана 1-31 мая 2007 г.

Структура и объем работы. Диссертационная работа состоит из введения, шести глав и заключения, изложенных на 137 страницах машинописного текста, содержит список литературы, включающий 30 наименований.

Содержание работы

В разделе «Введение» представлена оценка современного состояния разработок по теме магистерской диссертации, основание и исходные данные для разработки, обоснование необходимости проведения разработки, сведения о планируемом техническом уровне разработки и о результатах анализа аналогов и прототипов.

В первой главе производится описание предметной области и существующих проблем; производится выбор и обоснование критериев качества анализируемых систем, анализ аналогов и прототипов; формируется перечень задач, подлежащих решению в процессе исследования.

Во второй главе описаны подходы к построению жестомимического тезауруса и структура разрабатываемого мультимедийного тезауруса жестомимических и артикуляционных образов терминов информатики.

В ходе исследования были выделены следующие подходы к построению жестомимического тезауруса:

1. Построение жестомимического тезауруса на основе словесного тезауруса;
2. Построение жестомимического тезауруса на основе жестовой метрики;
3. Построение жестомимического тезауруса на основе смысловой метрики.

1. Построение жестомимического тезауруса на основе словесного тезауруса предполагает наличие словесного тезауруса: в словесный тезаурус добавляют жестовые обозначения соответствующих слов/словосочетаний, приводят жестовые примеры употребления данных слов/словосочетаний, словесное и нотационное описание жестов (рис.1).

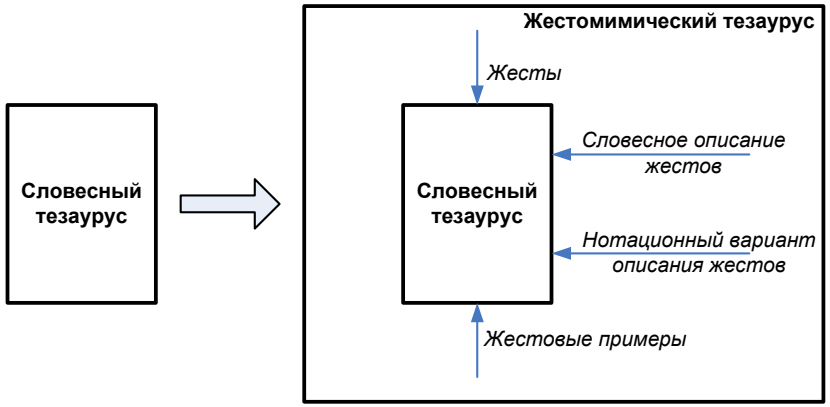


Рис.1 Построение жестомимического тезауруса на основе словесного

Рассматриваемый подход неприменим для целей создания общеязыкового тезауруса, поскольку разговорная жестовая речь, ввиду ее функциональной предназначенности и особенностей двигательной субстанции, помимо собственной лексики, характеризуется специфическими способами выражения смысла, отличными от способов, принятых в словесном языке.

В отличие от разговорной жестовой речи, калькирующая жестовая речь «калькирует» структуру предложений словесного языка и обладает

во многом схожими со словесным языком способами выражения смысла. Таким образом, учитывая функциональную предназначенность калькирующей жестовой речи, данный подход можно применять при построении терминологических жестомимических тезаурусов.

2. Построение жестомимического тезауруса на основе жестовой метрики предполагает представление жеста в виде упорядоченного набора составляющих его структурных единиц или структурных признаков жеста (см. рис.2).

Рис.2 Набор структурных признаков жеста

Таким образом, жест – это кортеж следующего вида:

$G =< e_1, e_2, ...e_K >$, где e_i – *i*-й структурный элемент жеста; К – число структурных единиц жеста.

Пусть имеется пространство жестов Ω, означивающих термины информатики (рис.3).

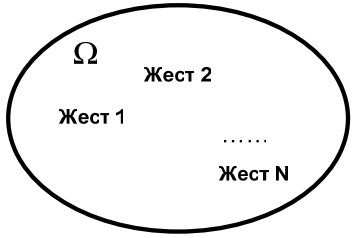


Рис.3 Пространство жестов, означивающих термины информатики

N – число жестов пространства Ω.

Поставим в соответствие каждым двум жестам a и b пространства Ω неотрицательное число ρ(a,b) (расстояние между a и b):

$$\rho(a,b) = \sum_{i=1}^{K} \left(e_{ai} - e_{bi}\right),$$

$$\left(e_{ai} - e_{bi}\right) = \begin{cases} 1, e_{ai} \neq e_{bi} \\ 0, e_{ai} = e_{bi} \end{cases},$$

где e_{ai} – *i*-й структурный элемент жеста a; e_{bi} – *i*-й структурный элемент жеста b; К – число структурных единиц жеста.

Можно показать, что для расстояния ρ выполняются следующие три условия [1]:

$\rho(a,b) = 0$ тогда и только тогда, когда $a=b$;

$\rho(a,b) = \rho(b,a)$ для любых двух жестов a и b из Ω («аксиома симметрии»).

$\rho(a,c) \leq \rho(a,b) + \rho(b,c)$ для любых трех жестов a, b и c из Ω («аксиома треугольника»).

Таким образом, расстояние ρ представляет собой метрику, а пространство Ω – метрическое пространство жестов.

Расстояние ρ равно числу структурных элементов, значения которых для рассматриваемых жестов не совпадают (рис.4). Значение ρ – изменяется от 0 до К; оно тем меньше, чем ближе данные жесты в заданной системе признаков. Значение ρ формально определяет силу связи между жестами пространства Ω.

Такой способ задания расстояния между жестами является аналогом «расстояния по Хэммингу».

Подсчитав значения ρ_{ij} для всех пар жестов пространства Ω, получим квадратную симметричную матрицу размером NxN, которую далее можно анализировать с помощью какого-либо метода автоматической классификации.

После того как выполнено разбиение жестов на группы и установлены связи между группами, тезаурусные статьи могут быть пополнены словесным описанием жеста, нотационным описанием жеста, жестовыми примерами и т.д.

Предложенный подход к конструированию жестомимического тезауруса позволяет установить формальные отношения между парами жестов пространства Ω. Если нас интересует изменение значений кон-

кретных структурных элементов жеста, то необходимо ввести систему весовых коэффициентов структурных признаков жеста (в рассмотренном нами варианте предполагалось, что все признаки имеют одинаковую степень важности).

Несмотря на наличие формальных связей между жестами пространства Ω, рассматриваемый подход (по аналогии с алфавитным и смысловым расположению слов в словаре [7]), представляет собой группировку жестов по «близости их звучания», т.е. является аналогом алфавитного расположения слов.

Недостатком данного подхода является отсутствие полного описания структуры жеста русского жестового языка.

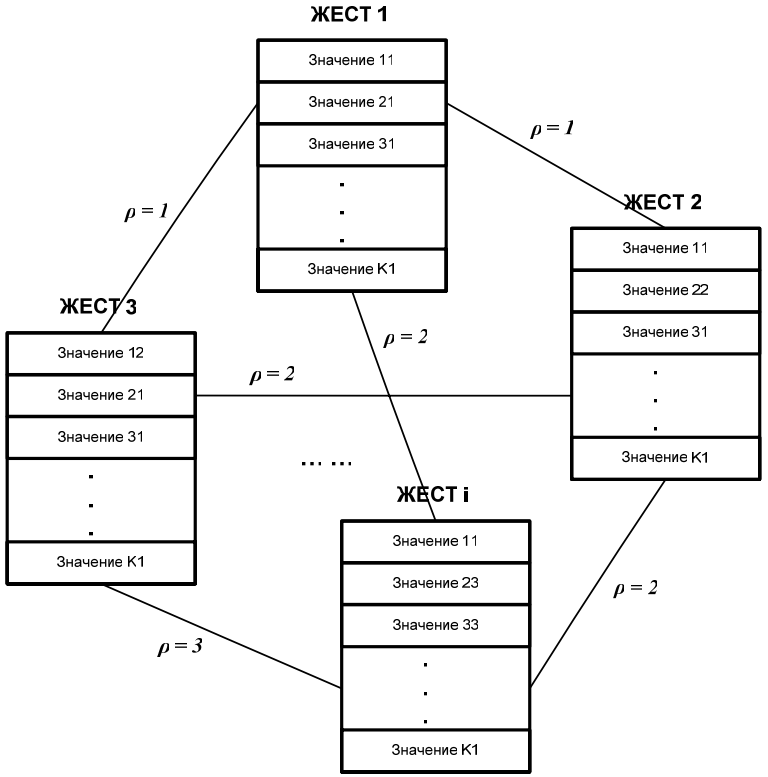


Рис.4 Связи между жестами пространства Ω

3. Построение жестомимического тезауруса на основе смысловой метрики предполагает представление жеста в виде упорядоченного набора смысловых признаков, в простейшем случае бинарных (рис.5). Аналогичный способ используется при описании слов (например, Ю.А. Шрейдер [9] выделяет порядка 20 признаков, описывающих слово).

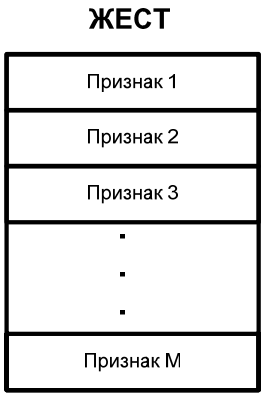


Рис.5 Набор смысловых признаков жеста

Таким образом, жест – это кортеж следующего вида:

$$G =< n_1, n_2, ...n_M >,$$

где n_i – i-й признак, описывающий жест; n_i может принимать значения 1 или 0 (например «большой» – 1, «маленький» – «0»); M – число смысловых признаков.

Имеется булево пространство жестов Ω, означивающих термины информатики (см. рис.3), где N – число жестов. Жест представляет собой булев вектор длиной M.

Расстояние между каждой парой булевых векторов пространства Ω можно вычислить по следующей формуле:

$$\rho_{ij} = \sum_{k=1}^{M} |a_{ik} - b_{ik}|.$$

Можно показать, что данное расстояние неотрицательно и для него выполняются условия 1-3, описанные выше. Расстояние, заданное таким образом, носит название «расстояния по Хеммингу». Пространство жестов Ω представляет собой метрическое булево пространство жестов.

Таким образом, расстояние ρ равно числу смысловых признаков, значения которых для рассматриваемых жестов не совпадают (см.

рис.4). Значение ρ – изменяется от 0 до M; значение ρ тем меньше, чем ближе данные жесты в заданной системе признаков.

Если нас интересуют конкретные смысловые признаки, то необходимо ввести систему весовых коэффициентов смысловых признаков жеста.

Подсчитав значения ρ_{ij} для всех пар жестов булева пространства Ω, получим квадратную симметричную матрицу размером NxN, которую далее можно анализировать с помощью какого-либо метода автоматической классификации.

После того как выполнено разбиение жестов на группы и установлены связи между группами, тезаурусные статьи могут быть пополнены словесным описанием жеста, нотационным описанием жеста, жестовыми примерами и т.д.

Недостатками данного подхода являются отсутствие полного лингвистического описания русского жестового языка, а также принципиальная сложность задачи выделения смысловых признаков, описывающих жест.

Ввиду специфики предметной области, разрабатываемый мультимедийный тезаурус жестомимических и артикуляционных образов терминов информатики строится на основе словесного тезауруса по предметной области «Информатика».

Проектирование структуры жестомимического тезауруса велось на основе структуры хорошо разработанного словесного тезауруса по Ю.Н. Караулову [6] и ГОСТ 7.25 «Тезаурус информационно-поисковый одноязычный. Правила разработки, структура, состав и форма представления». Структура словесного тезауруса по информатике была дополнена жестовыми обозначениями терминов, словесным и нотационным описанием жестов, жестовыми примерами.

1. Формирование классификационной схемы тезауруса

Совокупность понятий области «Информатика» была разделена на зоны в соответствии с тематическим делением терминов информатики в энциклопедическом словаре-справочнике по информатике Ф.С. Воройского [2]. Всего было выделено 6 зон и 37 понятийных групп.

Рубрики классификационной схемы тезауруса представлены в Приложении А магистерской диссертации.

2. Сбор массива лексических единиц и формирование словника тезауруса

Ввиду того что, время разработки жестомимического тезауруса было ограничено, сбор массива лексических единиц осуществлялся только

для рубрики 1.3 «Носители информации, документы, документация, издания».

Сбор лексики осуществлялся на основе энциклопедического словаря-справочника по информатике Ф.С. Воройского [2], ГОСТ 19781-90 (Обеспечение систем обработки информации программное. Термины и определения) и ГОСТ 28397-89 (Языки программирования. Термины и определения).

В словник тезауруса в соответствии с ГОСТ 7.25 могут входить следующие типы лексических единиц: одиночные слова (существительные, прилагательные, глаголы, наречия); именные словосочетания; лексически значимые компоненты сложных слов; сокращения слов и словосочетаний.

Требования к словосочетаниям и компонентам сложных слов, включаемых в словник, форме представления одиночных слов и аббревиатур представлены в ГОСТ 7.25.

В словник вошло порядка 100 терминов области «Информатика». Перечень терминов представлен в Приложении Б магистерской диссертации.

3. Формирование идеографической части тезауруса

При построении тезаурусных статей лексическим единицам приписываются соответствующие типы ссылок, отмечающие связь лексических единиц между собой. Типы и значения ссылок представлены в таблице 2.2 магистерской диссертации.

Неоднозначность значений лексических единиц в разрабатываемом тезаурусе устраняется путем указания связей между значениями лексических единиц, определяемых дефиницией.

Между лексическими единицами в соответствии с ГОСТ 7.25 устанавливаются отношения эквивалентности: лексические единицы объявляются эквивалентными, если замена одной лексической единицы на другую не приводит к изменению смысла текста.

В соответствии с ГОСТ 7.25 каждой лексической единице присваивается статус дескриптора или аскриптора.

4. Установление логико-семантических отношений между дескрипторами

Для дескрипторов устанавливаются парадигматические отношения, отражающие логико-семантические связи между понятиями, выражаемыми дескрипторами. Согласно ГОСТ 7.25 связь указывают путем внесения в дескрипторную статью ссылки, включающей обозначение согласно таблице ссылок (см. таблицу 2.2 магистерской диссертации) и связанный дескриптор.

5. Наполнение тезаурусной статьи жестовой информацией

Каждому термину тезауруса ставиться в соответствие набор жестов, означивающих данный термин. Жест сопровождается словесным и нотационным описанием (в главе 3 магистерской диссертации разработан вариант жестовой нотации). Кроме этого, приводятся примеры употребления данного жеста в калькирующей жестовой речи.

Таким образом, тезаурусная статья состоит из следующих элементов: наименование дескриптора; дефиниция; перечень всех «синонимов» дескриптора; перечень дескрипторов, связанных с данным дескриптором, с указанием типа связей; набор жестов, означивающих данный дескриптор; словесное описание жеста; нотационное описание жеста; примеры употребления жеста.

В тезаурусную статью неявно включены жестовые обозначения, словесное описание, нотационное описание и жестовые примеры «синонимов» дескриптора.

6. Формирование алфавитных указателей

Для мультимедийного тезауруса строится два типа указателей:

1) алфавитный указатель дескрипторов с адресом соответствующей зоны или зон. Например:

ДЕСКРИПТОР1 (2.1);

ДЕСКРИПТОР2 (3.4, 1.2).

2) алфавитный указатель слов (терминов и терминосочетаний) с адресами соответствующих дескрипторов. Например:

ТЕРМИН1 (ДЕСКРИПТОР2);

ТЕРМИН2 (ДЕСКРИПТОР2, ДЕСКРИПТОР4, ДЕСКРИПТОР10).

В третьей главе был произведен анализ структуры жеста в калькирующей жестовой речи и разработан вариант нотационного описания жеста.

1. Анализ структуры жеста

Анализ структуры жеста производился на основе набора жестов (среди которых были и жестовые обозначения терминов информатики), полученного в процессе опроса глухих и плохослышащих студентов МГТУ им. Н.Э. Баумана (студентам было предложено порядка 150 терминов области «Информатика», которые требовалось означить, придумывая собственный жест, или, пользуясь уже существующим жестовым обозначением).

Анализ полученных жестовых обозначений производился на основе визуального наблюдения за данными жестами в ходе просмотра соответствующих видеороликов, а также на основе уже существующих исследований структуры жеста.

Таким образом, были выделены следующие типы жестов: одноручные жесты; двуручные жесты (жесты с одинаковой формой рук; правая и левая руки выполняют одно и то же движение синхронно); жесты, исполняемые двумя руками.

Были выделены следующие структурные элементы жеста:

1. Конфигурация – положение и форма пальцев руки (рук);
2. Локализация – место исполнения жеста;
3. Движение.

Компонент *движение* состоит из следующих шести частей:

1. Часть руки, исполняющая движение (кисть, предплечье и т.д.);
2. Вид траектории движения (круговое, зигзагообразное и т.д.);
3. Ритмическая характеристика движения (повторяющееся, обратимое и т.д.);
4. Направление движения (вверх, вниз, от себя и т.д.);
5. Качество движения (непрерывное, прерывистое);
6. Плоскость, в которой выполняется движение.

Кроме этого, были введены вспомогательные модификаторы, позволяющие получать новые признаки компонента *конфигурация* на основе уже имеющегося набора признаков, уточнять место исполнения жеста, направление движения, способ касания.

В отличие от работ Г.Л. Зайцевой [5] и Л.С. Димскис [4], данное представление жеста позволяет описывать сложные жесты, например жесты, в которых изменяется конфигурация, вид траектории движения и т.д.

2. Разработка варианта нотационного описания жеста

Учитывая проблемную ориентированность данной работы, целью разрабатываемого варианта жестовой нотации является дальнейшее хранение и обработка символьной записи жеста в ЭВМ, что, прежде всего, повлияло на состав символов, используемых для описания признаков, характеризующих компоненты жеста.

Данную нотацию нецелесообразно использовать для скорописной фиксации жестовой информации.

Подробное описание жестовой нотации приведено в параграфе 2 магистерской диссертации.

В четвертой главе произведен выбор технологии реализации мультимедийного жестомимического тезауруса; разработаны схема взаимодействия модулей системы и схема взаимодействия пользователей с системой; приведено описание технологии фиксации и хранения жеста; разработан макет системы.

В пятой главе произведена оценка социального эффекта разработки и внедрения мультимедийного тезауруса жестомимических и артикуляционных образов терминов информатики. Рассчитана смета затрат на разработку, внедрение и эксплуатацию системы, рассчитан коэффициент социальной рентабельности проекта.

В результате произведенных расчетов, коэффициент социальной рентабельности оказался больше единицы, что подтверждает социальную эффективность проекта.

В шестой главе описана схема аутентификации пользователей с обеспечением защищённой передачи данных между клиентами и системой на основе протокола защищенного обмена данными SSL.

Литература

1. *Васильев Н.* Метрические пространства / Н. Васильев // Научно-популярный физико-матаматический журнал «Квант». – 1990. – N1. – С.16-23.
2. *Воройский Ф.С.* Информатика. Энциклопедический словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах / Ф.С. Воройский. – М.: ФИЗМАТЛИТ, 2006. – 768 с.
3. Группы специального назначения // Мир глухих. – 2006. – Май(№5).
4. *Димскис Л.С.* Изучаем жестовый язык: пособие для студентов дефектол. фак. / Л.С. Димскис. – Минск: «НМЦентр», 1998.
5. *Зайцева Г.Л. Жестовая речь.* Дактилология : учеб. для студентов вузов / Г.Л. Зайцева. – М.: «Владос», 2004. – 191 с.: ил.
6. *Караулов Ю.Н.* Лингвистическое конструирование и тезаурус литературного языка / Ю.Н. Караулов. – М.: Наука, 1981.
7. *Морковкин В.В.* Идеографические словари / В.В. Морковкин. – М.: Изд-во МГУ, 1970.
8. Нормативная база ГСНТИ [Электронный ресурс]. – Режим доступа: http://www.gsnti-norms.ru.
9. *Шрейдер Ю.А.* Сложные системы и космологические принципы / Ю.А. Шрейдер // Системные исследования. Ежегодник. – М.: Наука, 1971.

А.А.Кононков

ЭЛЕКТРОННЫЙ ДИАЛЕКТОЛОГИЧЕСКИЙ АТЛАС РУССКОГО ЯЗЫКА[8]

Основное содержание работы

Во введении обосновывается актуальность темы, формулируется цель работы, состав решаемых задач, приводится перечень основных результатов, и излагается краткое содержание глав.

В первой главе «Анализ методов и систем обработки и представления данных диалектологических исследований» приводятся основные сведения о диалектологии как науке, о диалекте, как главном предмете ее исследования. Рассматриваются автоматизированные системы, которые исследователи-диалектологи используют в своей работе. Также анализируются трудности, возникающие при обработке диалектологических данных, среди них следует отметить:

- Большое количество разнородных данных, вследствие чего, при их компьютерном представлении следует уделять повышенное внимание к организации хранения данных на машинном носителе.
- Необходимость наглядного представления данных и результатов исследований.

Анализируется проблема удобного и в той же мере достоверного отображения информации. Так даже при существовании визуального представления диалектологических исследований, например, Диалектологический Атлас Русского Языка, существует проблема отображение информации. Суть заключается в следующем – в силу особенностей исходного материала на одном листе атласа отображается только не-

[8] Статья представляет собой автореферат магистерской диссертации по специальности 230100 – «Информатика и вычислительная техника», выполненного на кафедре «Системы обработки информации и управления» Московского государственного технического университета им. Н.Э. Баумана в 2006 г. Научный руководитель к.т.н., доц. Ю.Н.Филиппович; консультант по экономике: к.т.н., доцент В.А.Дадонов; рецензент к.ф.н., старший научный сотрудник ИРЯ им.В.В.Виноградова РАН И.И.Исаев

сколько диалектологических явлений. Для своих исследований диалектологам необходимо комбинировать не только изображения явлений на одном листе, но также и на разных листах. Данная задача может быть решена созданием Электронного Диалектологического Атласа, разработанного с учетом пожеланий диалектологов по его функциональности и пользовательскому интерфейсу.

Далее рассматривается структура Диалектологического Атласа Русского Языка, который выходит с 1986 г. и является результатом работ исследователе Института Русского Языка Российской Академии Наук. Данная структура представлена на рис. 1.

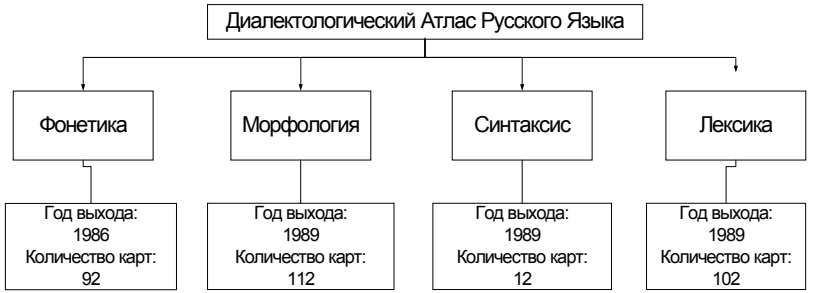


Рисунок 1.
Структура ДАРЯ с указанием количества карт в каждом выпуске.

Приводятся основные принципы построения систем для отображения картографической информации и сравнение систем автоматического районирования. Среди них можно выделить системы "ЭКОКАРТ" и GISCluster.

На основании проведенного анализа делается вывод о необходимости самостоятельной разработки системы для отображения результатов диалектологических исследований. Также отмечается, что наиболее рациональной будет разработка системы по модульной схеме, исходя из целого ряда причин – основная из которых, в том что эта схема позволяет начать тестирование части функциональности системы до конца разработки ее целиком, что ускоряет процесс создания.

Во второй части главы производится общий обзор методов группировки объектов по однородным группам на основании описывающих их признаков. Данный процесс называется классификация и является важным процессом в любом исследовании. Классификация дает возмож-

ность выявить набор схожих объектов исследования, что позволяет значительно упростить их анализ. Проблема выделения однородных групп рассматривается в статистике как задача группировки исходных данных. Принципиально существует два типа группировок:

Типологическая группировка – это разбиение совокупности на качественно однородные группы, характеризующие некоторые типы (классы) явлений, например группировка людей по полу или по используемому в их речи диалекту.

Структурная группировка – расчленение качественно однородной совокупности на группы, характеризующие строение совокупности, ее структуру. Фактически под структурой понимают распределение объектов по интервалам.

Главная цель кластерного анализа — нахождение групп схожих объектов в выборке данных. Эти группы удобно называть кластерами. Не существует общепринятого или просто полезного определения термина «кластер», и многие исследователи считают что отсутствует необходимость в поиске такое определение, Выделяют для рассмотрения достаточно много свойств которыми обладают кластеры, наиболее важными из которых являются: плотность, дисперсия, размеры, форма, отделимость кластеров друг от друга.

В работе способы определение и нахождение этих величины рассматривается более подробно.

Далее в главе показываются существующие алгоритмы кластеризации. Для наглядности деление методов можно представить графически на рис. 2.

Рисунок 2 Классификация методов кластерного анализа

Выделено семейство алгоритмов, которые могут использоваться для анализа данных диалектологических исследований. Итеративные автоматизированные методы использующие в качестве меры сходства таксономическое отношение.

В последнем параграфе главы дается постановка задач исследования как и в естественно языковой, так и в формальной форме.

Во второй главе «Методика машинного представления результатов диалектологических исследований» сначала предлагается методика оцифровки карт ДАРЯ, вторая часть главы посвящена вопросам формальной постановки задачи кластеризации говоров и рассмотрению имеющихся мер сходства для сравнения объектов описанных бинарными признаками.

По результатам материала первой части второй главы была разработана технология представления существующих карт ДАРЯ изображенная на рисунке 3.

В прямоугольниках жирным шрифтом представлены этапы методики. Под ними дана детализация этапов по процедурам. В конце каждого этапа приводится ожидаемые результаты операций.

Кратко остановимся на каждом из этапов методики:

*Сканирование***.** Задача сканирования – получить растровое изображение достаточного качества для дальнейшей успешной работы по векторизации карт. Очевидно, что чем больше разрешение сканирования и чем больше цветов имеет отсканированное изображение , тем больше файл и тем больше аппаратных ресурсов требуется, что бы с ним работать. Поэтому надо постараться оптимальное соотношение между качеством отсканированного изображения и размером файла. При сканировании листов Атласа было выбрано качество сканирования в 600 точек на дюйм и насыщенность в 16 миллионов цветов. Данные параметры подбираются путем нескольких пробных прогонов.

“Чистка” сканированных изображений. После сканирования получается изображение, которое, как правило, можно улучшить для того, чтобы векторизация была менее трудоемкой. Понятно, что в процессе векторизации программа сама определяет, где проводить векторную линию. При этом она ориентируется на пиксели растрового изображения. Предварительно в параметрах векторизации необходимо задать, какие разрывы в линиях векторизатору следует "не замечать". Чтобы растр был более аккуратный и имел меньше разрывов, точнее и ровнее сами линии, его необходимо специальным образом подготовить к векторизации. Набор инструментов для чистки растрового изображения зависит

от программного обеспечения. В программе которая использовалась при оцифровке карт Атласа, набор инструментов был достаточно широк, поэтому чистку изображения возможно было произвести на должном уровне, что позволило уменьшить количество ошибок при векторизации карт.

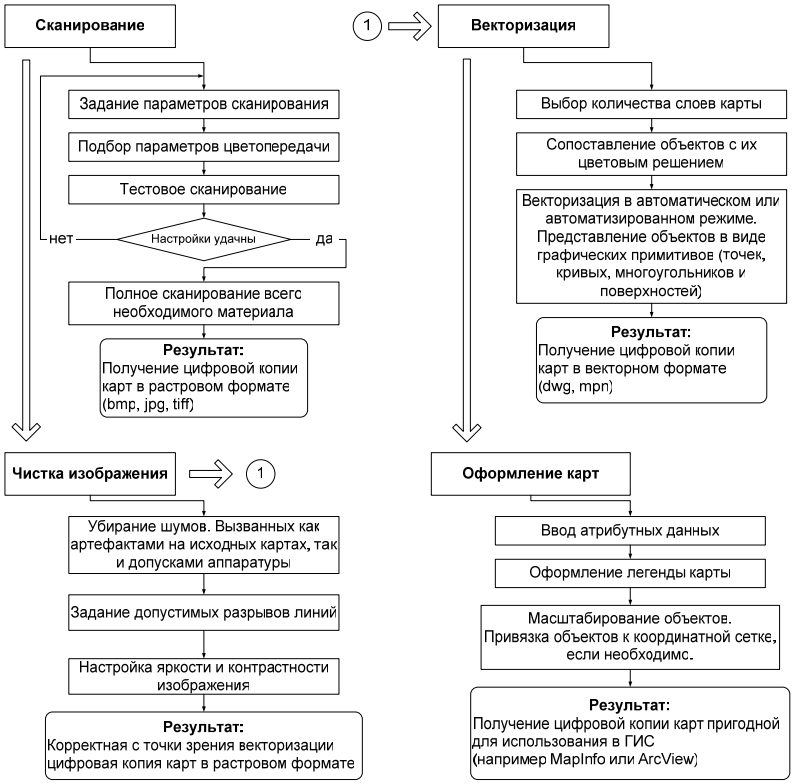


Рисунок 3 - Методика оцифровки карт ДАРЯ.

Векторизация или оцифровка. По сути это процесс превращения объектов на сканированной и калиброванной карте, полученной в результате выше описанных действий, в набор векторных объектов. То есть вместо набора черных и белых точек получается список геометрических характеристик: координаты вершин, цвет, толщина линии. Сканированное изображение становится "красивым" и масштабируемым, но

прежде всего этот процесс позволяет обеспечить возможность связи объектов на карте с их описанием в базе данных за счет присвоения так называемых пространственных ключей (то есть уникальных идентификаторов графических объектов, совпадающих с значениями специальных ключевых полей, описывающих эти объекты в базах данных).

Ввод атрибутивных данных. Атрибутивные данные - это сведения о том, что означает определенная точка на карте, например, село или город. А атрибутивная таблица- состоит из двух колонок: в одной колонке учтены все точки на карте, а в другой написано, что собой эта точка представляет. Естественно, что колонок может быть много, географические координаты, название и пр. Здесь необходимо было привязать к географическим пунктам те диалектологические признаки, которые там имеют место.

Оформление карт. Когда на экране открывается неоформленная карта, которая представляет только точки, линии и полигоны. Их можно раскрасить по имеющимся в атрибутивной таблице значениям атрибутов.

Во второй части главы строится формальная модель данных с которыми приходится иметь дело диалектологам и приводятся основные моменты исследований требующие автоматизации и применение кластерного анализа. Данные можно представить в виде $I = \{I_1, I_2, ..., I_S\}$,где множество I - множество населенных пунктов, со своими говорами; для Диалектологического Атласа Русского Языка S = 4195.

Также можно выделить множество *C* наблюдаемых в данном населенном пункте диалектологических явлений (признаков или характеристик), для любого пункта из множества *I*.

$C = \begin{Bmatrix} C_1 \\ C_2 \\ ... \\ C_p \end{Bmatrix}$, где *p* – общее количество признаков в исследовании, $p = 4416$;

Результат измерения *i*-той характеристики I_i объекта можно обозначить как x_{ij}, тогда вектор $X = [x_{ij}]$, размерностью $p \times 1$ будет отвечать каждому ряду измерений для *j*-го населенного пункта.

Таким образом, по результатам опросов на местности диалектологи для множества объектов *I* располагают множеством измерений $X = \{X_1, X_2, ..., X_s\}$, которые описывают множество I. В диалектологи-

ческих исследованиях измерения бинарные т.е. значения x_{ij} может принимать всего два значения 0 и 1 (1 – если признак присутствует в данном населенном пункте, 0 – если отсутствует).

На данный момент основным методом создание Диалектологических атласов является экспертная оценка множества признаков исследователями.

Рассмотрим простой пример. Допустим, перед исследователями стоит задача по результатам проведенных экспериментов выстроить разбиение говоров среднерусской части России на некоторое количество диалектологических групп. Стоит отметить, что русские говоры, в том числе и на территории ДАРЯ изучены не полностью. Диалектологи постоянно обнаруживают в них много нового: говоры развиваются, непрерывно меняют свой облик – в них исчезают одни и появляются другие языковые черты. Вместе с тем следует отметить то, что при всем многообразии диалектологических «находок» обнаруживается и их частный характер: они не являются характеристиками говоров больших территорий и, следовательно, не противопоставляют значительные массивы говоров. Поэтому диалектологи при разбиении говоров, на основании своих знаний и опыта выделяют для деления такие признаки, которые существенно различаются своим распределением по выделенным группам говоров. Далее диалектологии анализируют распределенные диалектологические исследований, затем наносят их на географическую карту. Получившиеся множество однородных точек обводит художник, так получаются листы Диалектологического Атласа Русского Языка.

Этот метод имеет следующие недостатки:

если формально описать задачу еще получается, то решение очень трудно поддается формализации из-за наличия большого влияния человеческого фактора.

большая размерность данных требующих анализа, матрица объектов и признаков получается размерностью 4000×4000, это ведет к большим умственным и временным затратам при анализе данных.

Для преодоления этих трудностей уместно использовать разработанные алгоритмы и методы кластерного анализа.

Далее рассматривается выбор мер сходства между объектами, приводятся основные из них, которые используются для измерения расстояния между говорами, информация о которых представлена в бинарном виде. Некоторые из них преведены в таблице 1. В данной таблице s_{ij} - значение меры близости между объектами *i* и *j*.

Таблица 1.
Меры сходства для определения близости между говорами.

Название	Формула
Мера близости Журавлева	$s_{ij} = \sum_{l=1}^{p} X_{ij}$
Мера близости Миркина	$s_{ij} = \sum_{l=1}^{p} q_{ij}$, Где q_{ij} - общее число совпадающих значений свойств.
Таксономическое отношение	$t_{ij} = \frac{1}{n} \sum_{k=1}^{n} w_r;$

Преимущество формулы таксономического отношения перед другими формулами определяющими меру близости между бинарным данным, определяется тем что, известные автору меры близости, в том числе и учитывающие тем или иным способом веса признаков, позволяют лишь сравнивать степень близости между различными объектами множества. Судить о том, могут или не могут какие-либо два объекта принадлежать к одной и той же классификационной единице, можно при помощи этих коэффициентов только на основании анализа полученных с их помощью значений величин попарной близости всех объектов рассматриваемой совокупности. Т.е. мы не можем определить пороговую близость сравниваемых объектов заранее – она устанавливается только после изучения объектов всего множества. Таксономическое же отношение дает ответ на принадлежность объекта множеству сразу.

В третьей главе «Применение методов кластерного анализа в диалектологических исследованиях» излагается суть метода кластеризации с использованием таксономического отношения, даются основные определения, а также формализовано описывается алгоритм, в дальнейшем использованный для разработки математического модуля Электронного Диалектологического Атласа Русского Языка

В начале главы даются некоторые рекомендации при подготовки данных для кластеризации, Матрица рассматриваемой совокупности объектов и описывающих их признаков должна удовлетворять следующим условиям:

отсутсвие полностью нулевых или заполненных единицами строк – что является первым признаком ошибки ввода, и такие строки следует удалить из рассмотрения;

также следует проверить матрицу на отсутствие значений, в какой либо из ее ячеек, эти элементы матрицы заполняются нулями. И считается, что если значение отсутствует, то исследуемое диалектологическое явление в данном населенном пункте не наблюдалось.

Далее вводится понятие таксономического отношения и даются формулы по которым оно вычисляется. Кратко это можно изложить следующим образом:

Предположим, если у нас имеется два объекта (говора) i и j, тогда таксономического отношение между двумя объектами будет представлять собой:

$$t_{ij} = \frac{1}{n}\sum_{k=1}^{n} w_r, \text{ где:} \qquad (1)$$

n – общее число признаков;

r – номер признака;

w_r – показатель близости между сравниваемыми объектами по признаку с номером r;

Мера близости w_r рассчитывается следующим образом:

$$w_r = \frac{\left(S - k_r\right)}{k_r}, \qquad (2)$$

если в обоих сравниваемых объектах признак с номером r присутствует;

$$w_r = \frac{k_r}{\left(S - k_r\right)}, \qquad (3)$$

если в обоих сравниваемых объектах признак с номером r отсутствует;

$$w_r = -1, \qquad (4)$$

если сравниваемые объекты по признаку r не совпадают: в одном из них этот признак присутствует, в другом - отсутствует.

В формулах (2), (3):

S – общее число рассматриваемых объектов.

k_r – число объектов, в которых признак r присутствует.

Преимуществом формулы таксономического отношения перед другими способами нахождения меры сходства между объектами заключается в том, что с помощью нее, вывод о принадлежности двух сравни-

ваемых объектов к одной группе (если, конечно, она выделяется) может быть сделан без анализа попарных мер сходства всех объектов множества:

при $t_{ij} < 0$ объекты не могут принадлежать одной группе;

при $t_{ij} \geq 0$ объекты могут принадлежат к одной группе;

Основное свойство таксономического отношения можно сформулировать следующим образом – чем больше значение этого отношения тем теснее связь между этими объектами.

Метод нахождения однородных групп с использованием таксономического отношения может быть описан следующими формальными этапами:

На первом этапе:

Для $\forall I_i, I_j \in I$, где $i, j = \overline{1..S}$, S – количество объектов в рассматриваемой совокупности.

Найти $t_{ij} = \frac{1}{n}\sum_{k=1}^{n} w_r$, где r – номер признака; w_r – показатель близости между сравниваемыми объектами по признаку с номером r;

И построить треугольную матрицу таксономических отношений:

$$T = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1n} \\ & t_{22} & & \dots \\ & & t_{n-1}t_{n-1} & t_{n-1n} \\ & & & 1 \end{bmatrix}$$

На втором этапе:

$\forall I_i, I_j \in I$, где $i, j = \overline{1..S}$, S – количество объектов в рассматриваемой совокупности.

Имея матрицу T найти множество $\{\mu_1 .... \mu_k\}$ (k – количество групп) не пересекающихся групп объектов, таких что $\forall I_i, I_j \in \mu_r, t_{ij} \geq 0$, $i, j = \overline{1..m}, m$ – количество объектов в группе r –той группе, $r = \overline{1..k}.$

Выше изложенный метод, впервые описанный Е.С.Смирновым[9], недостаточно устойчиво работает с данными большой размерности уже на

[9] Смирнов Е.С. Таксономический анализ,- М:Наука, 1969.- 356 с.

данных 100 объектов на 100 признаков время счета этого алгоритма достаточно велико.

Для увеличения скорости работы данного алгоритма вводится понятие эталона типичности. Это позволяет как бы выбрать центры кластеризации и на основании знания этих центов находить таксономическое отношение ни с каждым говором совокупности, а только с используемыми говорами. Данный метод был предложен Н.Н.Пшеничной[10] и дал хорошие результаты.

Эталон типичности (ЭТ) — это некий объект, характеризующий некоторую совокупность говоров по признакам, наиболее распространенным в данной совокупности. Наиболее распространенными считаются признаки, которые отмечены в половине или более говоров, т.е. относительная частота которых в совокупности $p \geq 0{,}5$. Эти признаки и образуют *эталон типичности* данной совокупности. Таким образом, *эталон типичности* какой-либо совокупности говоров - это комплекс признаков, наиболее распространенных в данной совокупности, или иначе говоря, это искусственно созданный (смоделированный) "говор", в характеристику которого входят только признаки, наиболее распространенные в данной совокупности (признаки, относительная частота которых в совокупности говоров, для которой моделируется *ЭТ*, $p \geq 0{,}5$, входят в ЭТ, а признаки с $p < 0{,}5$ в ЭТ не входят).

Данный метод может также может быть изложен в виде пошагового алгоритма:

На первом шаге эксперты-диалектологи на основании содержательного представления о говорах (на основании данных карт ДАРЯ) выделяют небольшую часть говоров, принадлежащих к одной однородной группе (какая именно группа пока не известно). Данную часть говоров называем эталоном типичности. Следует отметить, что в случае если выделение такого говора затруднительно, за исходный «шаблон» ведующей группы принимается какой-либо говор, который условно считаем что он и будет родоначальником будующей группы. Выделившуюся часть говоров будем называть группой нулевого приближения, а ее эталон типичности – ЭТ нулевого приближения.

На втором шаге вычисляется таксономическое отношение с J^0 каждого из говоров всей исследуемой совокупности, включая те говоры, которые вошли в группу нулевого приближения. Все говоры исходной совокупности, имеющие положительные таксономические отношения

[10] Пшеничнова Н.Н. Типология русских говоров.- М.:Наука,1996.- 207с.

(положительные связи) с J^0, образуют группу следующего, первого, приближения. Определяем *эталон типичности* группы говоров первого приближения J^1. Затем, вычисляем таксономическое отношение каждого говора исходной совокупности, включая и те, которые вошли в группу первого приближения, с J^1, и найдем группу второго приближения J^2 (она состоит из говоров, имеющих положительное таксономическое отношение с J^1). Далее аналогично определяются группы говоров и их *эталоны типичности* в последующих (третьем, четвертом и т.д.) приближениях.

Как видим второй шаг алгоритма представляет собой итерационный процесс, на каждом этапе которого предыдущая группа может как расширится за счет пополнения новыми говорами, так и уменьшится за счет выбывания из нее некоторой части говоров. Следует проверять количество пришедших и ушедших из группы говоров и пересчитывать эталон типичности для группы, Так же следует следить, чтобы говоры из эталона типичности по ошибки не были исключены из группы, для которой они являются родоначальниками. Такое в принципе должно быть не возможно, но в процессе работы алгоритма из-за некоторой приближенности проверок иногда случается.

Процесс по выделению групп стоит продолжать пока оставшаяся часть говоров, не вошедшая ни в одну группу, достаточно велика.

После окончания деления совокупности говоров на первом уровне, возможно, перейти к анализу каждой из единиц деления первого уровня отдельно. В результате выстроится некоторое иерархическое деление, которое может быть изучено более детально, нежели группы первого уровня деления. Получившиеся классификация будет индуктивной, т.е. ранг деления будет возрастать от общего к частному.

Важно отметить, что изложенный метод позволяет оценить и выделить пограничные говоры или объекты, что достаточно редко встречается в других методах кластеризации и позволяет строить пересекающиеся кластеры.

В четвертой главе «Описание системы электронный диалектологический атлас русского языка» рассмотрена общая архитектура созданной системы «Электронный диалектологический атлас», ее основные модули и структуру их взаимодействия. Также рассматривается функционал, который разбит на модули и структура входных и выходных данных.

Программа имеет модульную структуру. Модули разбиты по принципу выполняемых ими функций. Такими функциями являются:

Первичная загрузка данных из файлов, содержащих информацию о диалектологических исследованиях.

Модуль работы с оцифрованными картами Атласа. Он также содержит процедуры визуализации результатов кластеризации.

Математический модуль (содержит программную реализацию алгоритмов кластеризации).

Главный модуль программы. Осуществляет диалог с пользователем. Сюда же вынесена часть системы, которая осуществляет ведение базы данных.

Общая функциональная схема системы представлена на рисунке 4. Стрелочками указаны потоки данных циркулирующих в системе. Взаимодействие вспомогательных модулей системы происходит через основной модуль программы. Основной модуль программы обеспечивает диалог с пользователем, и передачу данных внутри системы.

Настройки системы хранятся в настроечном файле программы. В нем же хранятся пути к файлам системы, картам Атласа, формат входных файлов.

Программа оперирует несколькими типами входных файлов: графические (содержат визуальное отображение листов Атласа); текстовые (содержат информацию о листах Атласа в формализованном текстовом виде).

Первым типом текстовой информации являются файлы характеристических таблиц. Данные файлы были созданы при работе над системой АИС ДАРЯ, разработанной в 1987 году И.А.Исаевым. Данная программа функционировала на ЭВМ СМ-4 под управлением операционной системы ДОС-РВ. В системе были следующие файлы, представляющие интерес для создателей «Электронного диалектологического атласа». Информация, хранимая в этих файлах, описывает карты ДАРЯ.

Структуру информации, содержащейся в файле характеристической таблицы, можно представить в виде двумерной матрицы. Номер строки соответствует абсолютному номеру населенного пункта, номер столбца абсолютному номеру диалектологического признака. Каждый элемент этой матрицы равен либо 0, либо 1 и означает отсутствие или присутствие в изучаемом пункте данного признака (0 – отсутствие, 1 – наличие). С учетом общего количества пунктов. Далее в работе следует описание входных файлов системы.

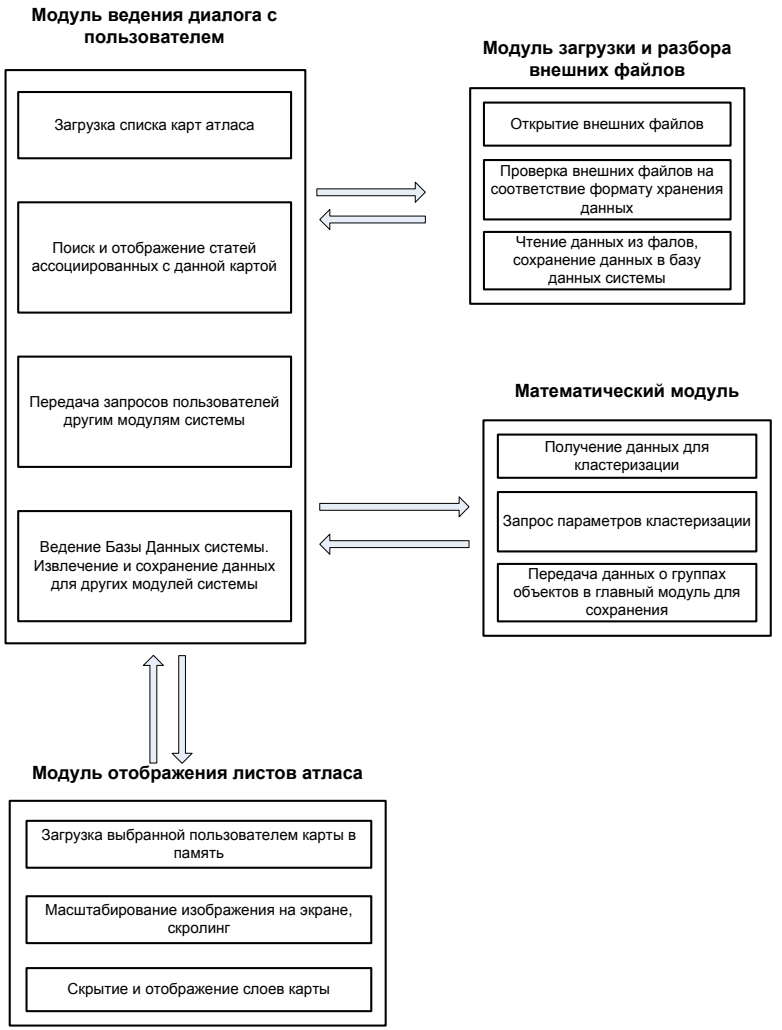


Рисунок 4 - Общая схема функционирования системы.

В пятой главе «Экономические аспекты проведения научного исследования» обосновывается, что разработка Электронного Диалектологического Атласа Русского Языка (ЭДАРЯ) не является коммерческой по целому ряду причин. Главная из них в том, что список пользо-

вателей ограничен исследователями русского языка, а именно учеными диалектологами. Людей занимающихся в этой области знаний не так много, поэтому ЭДАРЯ продукт узкоспециализированный и скорее всего не будет востребован непрофессионалами. Поэтому целесообразно считать социальный эффект от внедрения ЭДАРЯ. Он учитывается по специальной методике предложенной Шековой (инициалы) по расчету Коэффициент социальной рентабельности (SR). SR, у ЭДАРЯ равен 0,09, что по аналогии с другими экономическими коэффициентами может означать улучшение условий труда диалектологов в среднем на 9%. Стоит отметить что эта цифра может быть в несколько раз выше, т.к. для расчета брались минимальные значения и расчет социальной эффективности носит довольно приближенный характер.

Также в данной главе приводится смета затрат на проведение исследований, подсчитывается ее себестоимость, которая равна 77290 руб.

Также стоит отметить, что использование электронной версии Диалектологического Атласа Русского Языка позволит значительно расширить количество исследователей, которые получат доступ к материалам Атласа, что несравненно даст положительный эффект в долгосрочной перспективе.

В шестой главе «Защита информации» рассматриваются методы защиты от несанкционированного копирования информации на компакт дисках. Защите интеллектуальной собственности — программного обеспечения, данных и пр. — сегодня уделяется довольно много внимания. Т.к. предполагается распространение программного продукта Электронный Диалектологический Атлас на оптических дисках, что и обусловило материал рассмотренный в данной главе. Существует несколько подходов к защите информации на оптическом диске. Первый из них заключается в запрете запуска программы без определенного ключа (физического или программного). Вторым подходом является запрет на копирование дисков. Но ни один из подходов не дает 100% защиты.

Информацию, представленную в виде двоичного кода, можно беспрепятственно скопировать. Задача состоит в том, чтобы сделать простое копирование бесполезным — как, например, в случае, если работоспособность программы зависит от наличия некоего физического ключа. Из возможных способов защиты для Электронного Диалектологического Атласа был выбран метод использования стороннего программного продукта, который в себе уже содержал алгоритмы реализующие основные методы защиты компьютерных дисков. Проведен анализ основ-

ные программы представленные на рынке в настоящий момент. Так как разработка является некоммерческой, то большее внимание уделялось выбору бесплатно распространяемым программам. По совокупности характеристик выбор пал на программу TZcopyprotection 1.2.

В **Заключение** приводятся следующие научные результаты работы:

Проведен анализ методов нахождения структуры диалектологических данных исследований русского языка с применением кластерного анализа.

Разработан усовершенствованный алгоритм кластеризации с использованием таксономического отношения. Данный алгоритм был применен при создании математического модуля ЭДАРЯ.

Предложена технология получения электронной версии карт Диалектологического Атласа Русского Языка.

Разработан действующий макет программного комплекса Электронного Диалектологического Атласа реализующий следующие функции:

отображение листов ДАРЯ и навигацию по ним;

хранение и отображение, поясняющих карты ДАРЯ, материалов;

хранение и импорт данных о населенных пунктах и диалектологических явлениях имеющих силу в данных населенных пунктах;

Список публикаций по теме диссертации

1. *Кононков А.А.* Анализ применения методов кластерного анализа для выявлении структуры данных. Отчет по НИР за 9–10 семестры.

2. *Кононков А.А.* Особенности применения геоинформационных технологий в диалектологических исследованиях. Отчет по НИР за 11 семестр

3. *Кононков А.А.* Системы автоматизированного районирования / Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. Выпуск 7 // Сост. и ред. Ю.Н.Филипповича. — М.: Изд-во ООО «Эликс+», 2005. — с.125–129.

4. *Кононков А.А.* Применение методов кластерного анализа при проведении исследований / Интеллектуальные технологии и системы. Сборник учебно-методических работ и статей аспирантов и студентов. Выпуск 8 // Сост. и ред. Ю.Н.Филипповича. — М.: Изд-во НОК CLAIM, 2006. — С.59–64.

А.В.Сиренко

ЛИНГВОКУЛЬТУРНЫЙ ТЕЗАУРУС РУССКОГО ЯЗЫКА[11]

Общая характеристика работы

Актуальность темы. Одной из наиболее динамичных и востребованных разделов компьютерной лингвистики сегодня является семантический анализ текстов. Ему есть множество применений в сфере образования, интернет-технологиях и во многих других областях, где есть необходимость в автоматизации обработки больших массивов текстовых данных. Семантический анализ – процесс выявления смыслового содержания слов и словосочетаний в предложении. Он обеспечивает нормализацию синтаксической структуры предложений, распознавание терминов, классификацию терминов по семантическим признакам, выявление определений терминов.

Тема работы «Лингвокультурный тезаурус русского языка» непосредственно связана с задачей семантического анализа текста. Применяемая в работе пятикомпонентная структура единицы языкового сознания была разработана чл. корр. РАН Карауловым Юрием Николаевичем. Также в работе используются результаты ассоциативного эксперимента в виде ассоциативной сети.

Дипломная работа выполнена в рамках проекта РФФИ 05-06-80284 «Языковое сознание нашего современника: когнитивная структура и лингвокультурное содержание».

Цель работы. Целью работы является разработка лингвокультурного тезауруса русского языка на основе баз данных фигур знания и ассоциативного эксперимента, а также моделирование пассивного режима ра-

---

11 Статья представляет собой автореферат дипломного проекта по специальности 230102 – «Автоматизированные системы обработки информации и управления», выполненного на кафедре «Системы обработки информации и управления» Московского государственного технического университета им. Н.Э. Баумана в 2007 г. Руководитель проекта к.т.н., доц. Ю.Н.Филиппович; консультанты: Г.Б. Полаева, к.т.н., доцент И.В. Переездчиков; рецензент к.т.н., доцент В.Н. Поляков.

боты когнайзера через нахождение путей между знаками в ассоциативной сети.

Задачи. Для достижения поставленной цели решаются следующие задачи:

1. Анализ предметной области работы.
2. Постановка задач, решаемых системой.
3. Разработка архитектуры системы
4. Проектирование базы данных лингвокультурного тезауруса.
5. Разработка алгоритма функционирования пассивного режима когнайзера
6. Проектирование алгоритмов взаимодействия с пользователем и экранных форм.
7. Реализация программного продукта.

Практическая ценность. В работе был разработан программный продукт, осуществляющий работу с базой данных фигур знания, выборку фигур по заданным критериям, добавление данных в тезаурус из внешнего источника, а также реализующий пассивный режим работы когнайзера. Работа выполнена в рамках гранта Российского фонда фундаментальных исследований.

Объем работы. Дипломная работа содержит 120 страниц, 92 рисунков и таблиц, 11 источников, 19 страниц приложений.

Основное содержание работы

Во введении производится общая постановка задач, указывается на возможности их решения.

Глава 1 содержит описание предметной области.

Язык рассматривается в качестве отражения представлений об окружающем мире внутри сознания субъекта, так называемого Языкового Сознания.

Языковое Сознание можно представить в виде структуры:

<Языковое Сознание> = <Единица Знаний о Мире> + <Языковая Единица>

В этой зависимости языковое сознание представляет собой когнайзер, в котором происходит постоянное преобразование Единиц Знаний о Мире в Языковые Единицы и наоборот. Работу когнайзера по переходу от слова (знака) к знанию будем называть активным режимом работы, а от знания к знаку – пассивным.

Разработке лингвокультурного тезауруса предшествовал ассоциативный эксперимент. В нем респонденту называлось некое слово (стимул) и его задачей было назвать слово, ассоциирующееся со стимулом.

В результате формируется пара слов «стимул-реакция», которые можно рассматривать в качестве отражения Языкового Сознания, где стимул выступает в роли Языковой Единицы, а реакция в роли Единицы Знаний о Мире.

Знания об окружающем мире можно представить в виде так называемых Фигур Знания. Это элементарные когнитивные единицы, которыми мы оперируем в повседневной жизни. Они содержат как интенсиональные характеристики, а именно Знак, Способ, Формула смысла, так и экстенсиональные, отражающие положение Фигуры знания в общем пространстве знаний. Такими параметрами являются когнитивная область и функция. Схематичное представление фигуры знания можно увидеть на рисунке ниже.

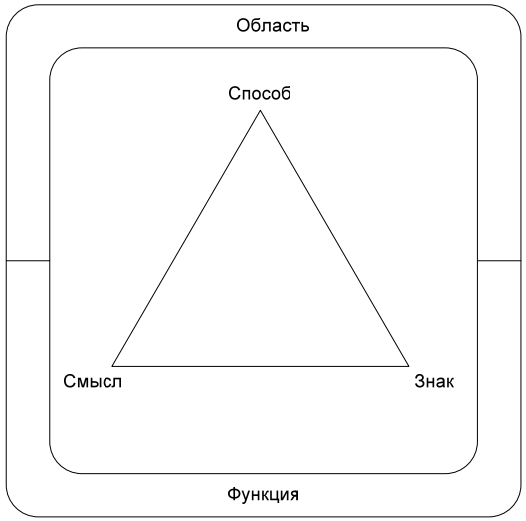


Рисунок 1 Фигура знания

В пассивном режиме работы когнайзера изначально имеется Формула смысла и, определенный через нее, Способ. Имея эти два параметра, когнайзер переходит к Знаку.

Фигуры знания в системе представлены материалами кроссвордов различной тематики.

Проводится обзор аналогов и прототипов. В качестве таковых приводятся системы:

- RSO Semantic Network SDK – инструментарий разработчика.

- www.lexfn.com – интернет-ресурс, работающий с англоязычной ассоциативной сетью.
- TextAnalyst – персональная система автоматического анализа текста.

Lexfn.com (интернет-ресурс компании «Datamuse Corporation», США, посвященный ассоциативным сетям) наиболее близок из всех представленных в обзоре систем к разработанной системе с точки зрения использования ассоциативной сети. Сайт предоставляет широкие возможности по поиску путей между знаками в ассоциативной сети. Система различает множество типов связей между словами, выполняет поиск не только непосредственно достижимых из начальной вершин сети, но и находит пути с ограничением по длине. На рисунке ниже изображен пример результата запроса.

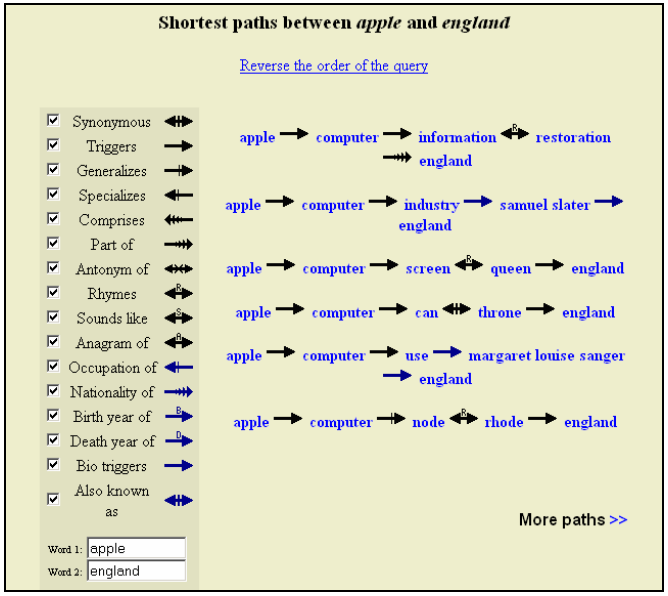


Рисунок 2 Результаты запроса

TextAnalyst разработан компанией “Microsystems, Ltd” в качестве инструмента для анализа содержания текстов, смыслового поиска информации, формирования электронных архивов. В результате анализа текста формируется семантическая сеть - основная структура, характеризующая смысл текста, в которой понятия (слова и словосочетания) объ-

единяются ассоциативными связями в соответствии с их совместной встречаемостью. Затем создается так называемое тематическое древо - представление структуры текста в виде многоуровневой иерархии тем и раскрывающих их подтем.

Библиотека *RCO Semantic Network*, разработанная компанией "Гарант-Парк-Интернет" позволяет автоматически анализировать содержание текстовых документов, представляя его в форме ассоциативной семантической сети. Ассоциативная семантическая сеть в программе представляет собой ориентированный граф, вершинами которого служат значимые темы, выделенные в анализируемом тексте, а дугами – связи между ними. С каждой вершиной связаны вес (значимость) и частота упоминания темы, а с каждой дугой – вес (сила) связи и частота подкрепления связи в тексте.

В главе 2 осуществляется разработка алгоритма пассивного режима когнайзера.

Для моделирования пассивного режима работы когнайзера необходимо осуществлять переход от формулы смысла к знаку, используя базу данных ассоциативного эксперимента. Для этого формула смысла разделяется на словоформы, для которых впоследствии ведется поиск соответствующих знаков ассоциативной сети.

Поиск путей от формулы смысла к знаку разделяется на поиск путей от каждого знака, входящего в формулу смысла к знаку фигуры знания.

К алгоритму поиска предъявляются требования:

- Определять длину пути.
- Восстанавливать сам путь при его нахождении.
- Должен быть не ресурсоемким.

Примененный в ЛКТ алгоритм поиска основан на алгоритме Дейкстра с единичной длинной пути (поиск в глубину).

Для каждой вершины в процессе поиска заполняется структура следующего вида:

Таблица 1

Наименование поля	Тип	Описание
Name	строка	наименование вершины
Level	целое число	Удаленность вершины от начальной
Old_elements	массив строк	Вершины, присутствующие в пути к данной вершине

Алгоритм поиска выглядит следующим образом:
1. Для начальной вершины заполняется структура.
Уровень = 0.
Список Old_elements пуст.
Текущей вершиной является начальная.
2. Если Level текущей вершины = Maxlevel тогда переход к пункту 8.
Иначе переход к пункту 3.
3. Рассматриваем список вершин, в которые можно попасть из текущей и которые не присутствуют в списке Old_elements текущей вершины.
4. Если в списке присутствует вершина, путь к которой мы ищем, выводим путь до текущей вершины с конечной вершиной как искомый путь. Обработка текущей вершины завершена. Переход к пункту 7.
5. Если список пуст, переход к пункту 8.
Иначе переход к пункту 6.
6. Для каждой вершины в списке заполняем структуру
Level = Уровень текущей вершины+1;
Old_element = Old_element текущей вершины + Name текущей вершины.
Для каждой вершины начинаем работу алгоритма в пункте 2.
7. Если Level < Minlevel переход к пункту 8.
Иначе вывод списка Old_elements в качестве пути до текущей вершины.
Переход к пункту 8.
8. Выход.

Функционирование алгоритма представлено на примере участка ассоциативной сети и поиске путей от вершины 1 к вершине 7.

Каждая вершина графа может быть точкой ветвления вычислений на n-1. Таких вершин может быть n-1. В результате сложность алгоритма вырастает до $\Theta(n^4)$.

Для оценки сложности алгоритма с учетом особенностей ассоциативной сети был проведен ее анализ.

Анализ ассоциативной сети, а также данные по аналогичным англоязычным сетям показал, что в рассмотрении путей длинной более 5 нет необходимости в силу высокой степени связности сети. Если вершина не достигается за 5 шагов, она считается недостижимой.

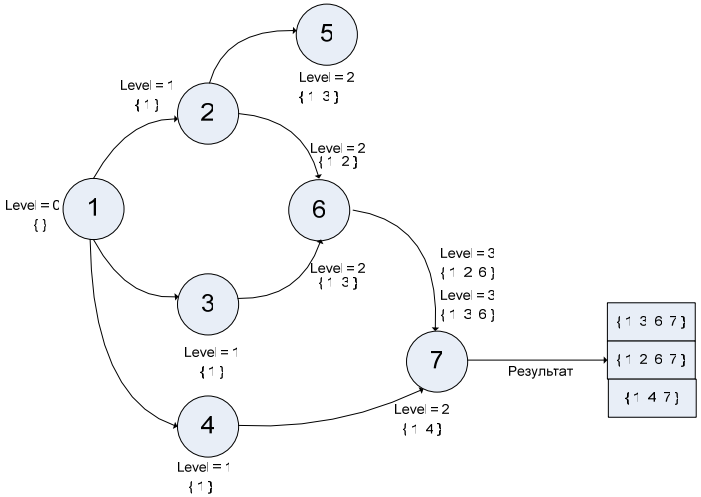


Рисунок 3

Для изучения функционирования поиска в ассоциативной сети было проведено изучение свойств сети в двух случаях:

- Изменение количества стимулов при неизменном числе реакций каждого стимула.
- Изменение количества связей сети, в том числе и с добавлением новых стимулов.

Для оценки характеристик сети при нарастании числа стимулов были произведены сокращения числа стимулов и расчет параметров:

- Количество связей
- Количество нелистьевых связей
- Процентное соотношение нелистьевых связей

Листьевыми будем называть связи, реакция в которых не является стимулов в каких-либо связях.

Таблица 2 Изменение числа стимулов

Процент стимулов	Количество стимулов	Количество связей	Количество нелистьевых связей	Процент нелистьевых связей
10	658	11005	701	6,36
20	1316	23788	3172	13,33
30	1974	36490	7328	20,08

40	2632	48560	12533	25,80
50	3290	60918	19649	32,25
60	3948	74305	28746	38,68
70	4606	86720	38630	44,54
80	5263	97857	49792	50,88
90	5920	109044	62322	57,15
100	6577	122059	78086	63,97

Ниже показана графическая интерпретация данной таблицы.

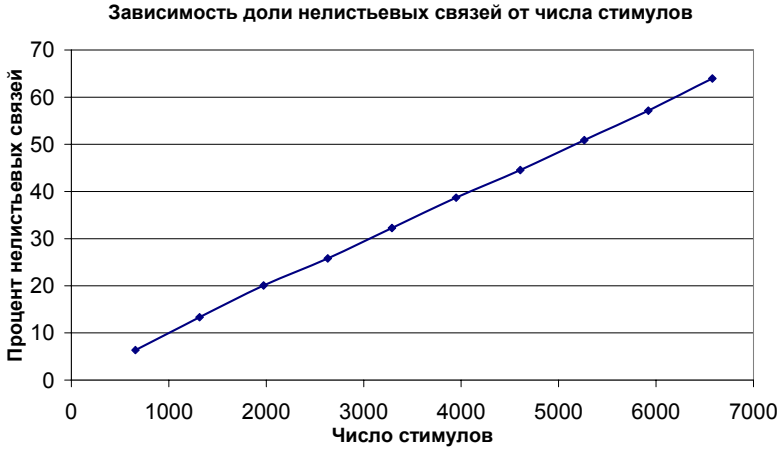


Рисунок 4. Изменение числа стимулов

На графике мы видим почти линейную зависимость процента нелистьевых связей от числа стимулов в диапазоне от 20 до 100 процентов стимулов исходной базы данных. Предложив респондентам новые стимулы, мы можем уменьшить число листьевых элементов сети.

Для оценки характеристик сети при нарастании числа связей были произведены сокращения числа связей и расчет параметров:

- Количество связей
- Количество стимулов
- Количество нелистьевых связей
- Процентное соотношение нелистьевых связей

На таблице ниже представлены результаты расчетов.

Таблица 3 Изменение числа связей

Процент связей	Количество стимулов	Количество связей	Нелистьевые элементы	Процент нелистьевых связей
1	932	1221	366	0,2997543
2	1521	2442	1013	0,414823915
3	1973	3663	1717	0,468741469
4	2343	4884	2491	0,51003276
5	2658	6105	3259	0,533824734
6	2929	7326	4009	0,547229047
7	3176	8547	4772	0,558324558
10	3838	12206	6934	0,568081272
20	5117	24412	14777	0,605317057
30	5740	36618	22706	0,620077557
40	6073	48824	30547	0,625655415
100	6577	122059	78086	0,639739798

На рисунках ниже приведены графические интерпретации содержимого таблицы.

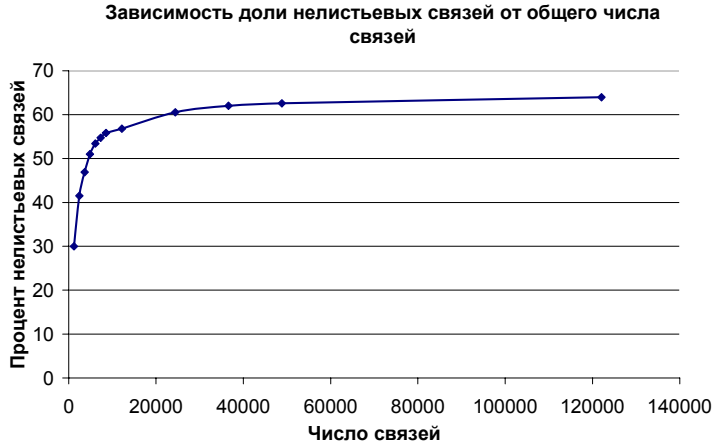


Рисунок 5 Зависимость доли нелистьевых связей

Судя по графику, при данном наборе стимулов, дальнейший опрос респондентов не приведет к значительному снижению доли листьевых элементов сети.

На графике мы видим заполнение ассоциативной сети во время опроса респондентов с точки зрения появления в ней новых стимулов.

Учитывая, что пути длиной более 5 в системе не рассматриваются, а также ограниченность количества реакций на стимул (что позволяет нам считать линейной зависимость между количеством вершин сети и ее связей с точки зрения расчета сложности алгоритма), результирующая сложность составляет $\Theta(k*n^2)$, где k - константа, k<<n.

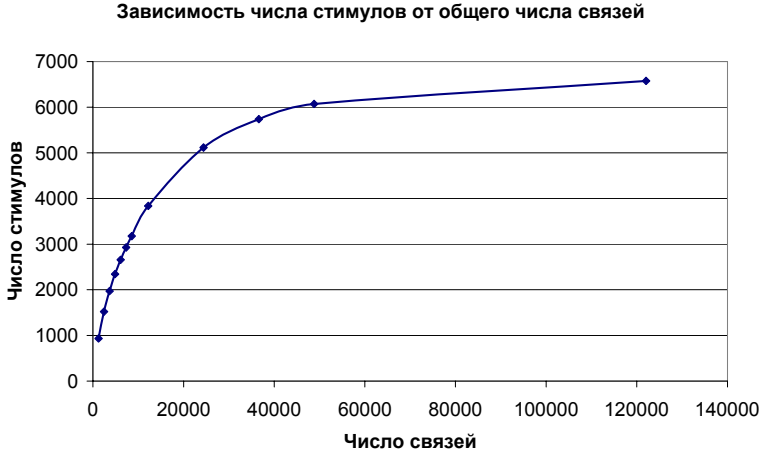


Рисунок 6 Зависимость числа стимулов от числа связей

Приведенный алгоритм обладает низкими требованиями к емкости используемой памяти, так как:

- для обработки вершин используется вызов рекурсивной функции
- поиск осуществляется в глубину
- после завершения работы экземпляра функции происходит освобождение памяти, занимаемой ею.

Приводится алгоритм доступа к базе данных Microsoft Access в случае изменения данных, а также поиска записей и их обработки.

Показан порядок формирования SQL-запроса, осуществляющего выборку фигур знания по критериям, установленным пользователем.

В главе 3 рассматриваются требования программному изделию, и в соответствии с ними обосновывается выбор Microsoft Access как используемой СУБД, а также Borland Delphi в качестве среды разработки.

Выполняется разработка схемы базы данных. Определяются функциональные зависимости схемы базы данных. Затем на их основе осуществляется доказательство оптимальности полученной схемы.

Производится выбор способа доступа к базе данных Access. В качестве альтернатив рассматриваются:

- Доступ через ODBC
- Компоненты DAO
- Технология ADO
- Компоненты KADao

ODBC (Open Database connectivity) – открытое соединение с базами данных – комплекс элементов, обеспечивающих стандартных средств взаимодействия с источниками данных при помощи синтаксиса SQL.

DAO (Data Access Objects) — объекты для доступа к данным — технология доступа к данным компании Microsoft, разработанная для работы с СУБД Microsoft Jet и позволяющая обращаться к базе данных Access.

ActiveX Data Objects — интерфейс программирования приложений для доступа к данным, разработанный компанией Microsoft и основанный на технологии компонентов ActiveX

KADao – компоненты доступа с базе данных Microsoft Access, использующие технологию DAO. Имеет простую систему классов, в значительной степени соответствующих таковой в ADO.

В результате выбор делается в пользу компонент KADao, имеющих простой для разработчика механизм работы, а множество функциональных возможностей.

Показан механизм импорта фигур знания в тезаурус на примере переноса данных их файла Microsoft Excel. Описана процедура обработки импортируемой таблицы, содержащей данные о добавляемых фигурах знания, для внесения их в схему базы данных тезауруса в виде фигур.

Приведены этапы осуществления пассивного режима работы когнайзера с точки зрения пользователя.

Дано описание справочной системы тезауруса и способа обращения к ней.

Глава 4 начинается с описания процесса установки тезауруса на компьютер пользователя.

Затем приводится граф диалога пользователя, отражающий возможные действия и доступ к основным функциям системы.

В главе присутствует описание методологии редактирования фигур знания. Это возможно двумя путями:

- Используя интерфейс главной формы, основанный на пятикомпонентной структуре фигуры знания
- Используя редактирование в табличном режиме

Табличный режим редактирования предназначен для пользователей, которым удобнее работать с данными в виде таблицы, чем со структурированным представлением фигур главной формы. Табличный режим редактирования не позволяет редактировать таблицы областей и способов.

Описан процесс выборки фигур знания по критерию. Необходимые действия для запуска данного режима работы тезауруса, настройки выборки, а также форма представления результата.

Показана работа пассивного режима когнайзера на примере фигуры знания

<Абориген – Коренной житель страны>

Производится поиск знаков ассоциативной сети, представленных в формуле смысла:

- Коренной
- Житель
- Страна

На поиск накладываются ограничения по длине путей: от 1 до 3 дуг.

В результате расчета найдены пути:

1. коренной-> абориген
2. житель-> абориген
3. страна->нищий->студент->абориген

В главе 5 рассмотрены вопросы эргономики и охраны труда оператора ПЭВМ**.** Показаны основные факторы, влияющие на утомляемость оператора. Приведены значения времени регламентированных перерывов в работе согласно СанПиН 2.2.2/2.4-1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Описаны эргономические характеристики рабочего места оператора, схема расположения отдельных элементов, а также психологические требования к рабочему месту.

Произведен расчет освещения рабочего помещения, определено необходимое осветительное оборудование и места его расположения.

Глава 6 содержит организационно-экономическое обоснование разработки.

Разработка системы «Лингвокультурный тезаурус русского языка» состоит из семи стадий:

1) Формирование требований
2) Разработка концепции
3) Техническое задание
4) Эскизный проект
5) Технический проект
6) Рабочая документация
7) Ввод в эксплуатацию

Производится расчет стоимости разработки системы

Таблица 4. Затраты на разработку проекта

Статья затрат	**Стоимость, руб.**
1. Оплата машинного времени	2480
2. Затраты на материалы	2550
3. Затраты на оплату труда – всего	54518
3.1. Основная заработная плата разработчика	36000
3.2. Дополнительная заработная плата	7200
3.3. Единый социальный налог (26%)	11232
3.4. Отчисления в фонд обязательного социального страхования от несчастных случаев на производстве (0,2%)	86
4. Накладные расходы	17280
5. Услуги сторонних организаций	1100
Общая сумма затрат на разработку проекта	77928

В качестве обоснования разработки приводится актуальность работы и создание программного продукта в рамках гранта РФФИ.

Заключение содержит выводы по результатам работы системы, а также пути возможных дальнейших исследований.

Заключение

Основные выводы и результаты работы:

В рамках дипломного проектирования была разработана схема базы данных тезауруса и доказана ее оптимальность

1. Был разработан алгоритм функционирования пассивного режима работы когнайзера, а также произведена оценка его сложности.
2. Произведен анализ структуры ассоциативной сети тезауруса, моделирующий процесс ее заполнения и сделаны выводы, касающиеся достаточной полноты опроса респондентов.
3. Разработана методика работы с базой данных фигур знания, включающая:
   - Поиск фигур по критерию
   - Редактирование фигур
   - Добавление фигур знания из внешнего источника
4. Произведен расчет эргономических параметров рабочего места оператора.
5. Определены затраты на разработку системы.

Методические работы

А.Ю.Филиппович

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ДИСЦИПЛИНЕ «АРХИТЕКТУРА АСОИУ»

Содержание дисциплины

Материалы дисциплины разделены на 8 модулей, каждый из которых содержит вопросы, рассматриваемые на лекциях или практических занятиях.

Модуль 1. Основные понятия и термины АСОИУ

Определение системы. Свойства системы: итерационность, ограниченность, эмерджентность, целостность, модельность и единство цели елементов.

Определение информации по Шеннону и Амосову. Формы выражения и представления информации. Информационные технологии. Обработка информации и управление.

Автоматизированная система. АСУ, САПР, АСНИ, АСУТП, АСУП, ИАС. Отличие АСУ от САУ. Функциональная и обеспечивающая часть АС. Виды обеспечения АС: организационное, правовое, методическое, техническое, математическое, программное, информационное, лингвистическое, эргономическое. КСА, АРМ и персонал АС. Компонент АС, программно-технический комплекс, комплектующие и программные изделия, информационное средство и изделие, информационная база.

Функциональная часть АС. Способы формирования функциональных подсистем: по функциям, фазам, объектам управления, бизнес–процессам. Сравнение достоинств и недостатков.

Основные принципы построения АСУ: системного подхода, принцип новых задач, первого руководителя, перманентности (непрерывного развития системы), разумной типизации проекта, автоматизации документооборота, единой базы, однократности ввода и многократности использования информации, комплексности задач и рабочих программ,

согласованности пропускных способностей различных элементов системы.

Архитектура АСОИУ. Архитектура как каркас и совокупность принципов построения АС. Архитектура как искусство проектирования и создания ИС и АС.

Жизненный цикл ИС. Модели ЖЦ: каскадная (водопадная), каскадная с промежуточным контролем, спиральная.

Стандарты в области ИТ. Назначение стандартов и область применения. Правила использования стандартов. Стандарты "де-факто" и "де-юре". Корпоративные стандарты. Классификация стандартов. Стандарты на организацию ЖЦ. Модели разработки. Российские стандарты на ИТ.

Модуль 2. Современные АСУП

АИС документального и фактографического типа. Возникновение БД и информационно-поисковых систем. Автоматизация предприятий. Классификация уровней управления предприятием по временному промежутку. Современная тенденция сокращения времени производственного цикла. Финансово-управленческие и производственные системы.

Системы промышленной автоматизации (САПР). Хронология развития. 2D и 3D системы, полная цифровая модель изделия, единое информационное пространство. CAD, CAE, CAM, CAPP, PLM и PDM системы. Схема функционирования PDM-системы и ее взаимосвязь с другими АС. MES, EAM и HRM системы.

Организационные (финансово-управленческие) системы. Классификация ИС по степени интеграции, сложности внедрения, функциональной полноте и стоимости. Примеры систем. MPS, MRP, CRP, FRP, MRP II, ERP, CSRP, SCM, CRM, ERP II системы. Недостатки "лоскутной" автоматизации.

Интеграция приложений. Переход к парадигме экосистемы бизнес–приложений. Отечественные аналоги — ОАСУ, ГАС. Подходы к интеграции: EAI (A2A), B2B, BPM-системы. Причины перехода к BPM-системам. Системы документооборота и потока работ (Docflow, Workflow). Недостатки жесткой унификации ERP систем. Языки описания бизнес–процессов и взаимодействия бизнес–приложений BPEL, BPML, BPMN.

Распределенные технологии создания ИС. Сервисно-ориентированная архитектура SOA и ее компоненты: ESB, BAM, AST. Событийно-ориентированная архитектура EDA. Теория обработки сложных событий.

Модуль 3. Комплексы стандартов 19-ой и 34-ой серий на АС

Перечень основных ГОСТ на АС. Назначение и общие положения.

ГОСТ 34.601-90. "Автоматизированные системы. Стадии создания". Назначение, общие положения, стадии и этапы создания АС: формирование требований, разработка концепции, техническое здание, эскизный проект, технический проект, рабочий проект (рабочая документация), ввод в действие (опытная эксплуатация, сопровождение). Общее содержание работ этапов создания АС. Перечень организаций, участвующих в работах по созданию АС.

ГОСТ 34.602-89. "Техническое задание на создание АС". Общие положения, состав и содержание, правила оформления.

Назначение, область распространения, состав, классификация и обозначение стандартов ЕСПД. ГОСТ 19.001, 19.004, 19.102 (сравнение с ГОСТ 34.601), 19.103, 19.105, 19.201 (сравнение с ГОСТ 34.602), 19.603 (общие требования). Назначения и отличия документов, определяемых стандартами 19.202, 19.301, 19.401, 19.402, 19.404, 19.502, 19.503, 19.504, 19.505.

ГОСТ 34.201-89 Виды, комплектность и обозначение документов при создании АС. Проектно-сметная и рабочая документация на АС. Комплектность документации. Общие сведения о видах и наименованиях документов.

РД 50-34.698-90 АС. Требования к содержанию документов. Соответствие документов и их содержания типовым моделям ARIS.

Модуль 4. Комплекс стандартов ГОСТ Р ИСО'МЭК на разработку программного обеспечения

Перечень основных ГОСТ Р ИСО'МЭК на разработку программного обеспечения, их назначение и область применения.

ГОСТ Р ИСО'МЭК 12207-99 - ИТ. Процессы жизненного цикла программных средств. Структура стандарта. Процессный подход. Участники процессов. Основные, вспомогательные и организационные группы процессов, состав работ и примерное содержание задач. Процессы с ключевых точек зрения (приложение С).

ГОСТ Р ИСО 9127-94. Системы обработки информации. Документация пользователя и информация на упаковке для потребительских программных пакетов. Сравнение с ЕСПД.

Сравнение ГОСТ Р ИСО'МЭК 12207, Rational Unified Process, ГОСТ 34 и 19 серий.

Модуль 5. ARIS — Архитектура методов разработки и моделирования ИС

ARIS — Ракурсы (View) ИС: организационный, управленческий (control), функциональный, информационный (data) и ресурсный. Трехфазная модель ЖЦ. Взаимосвязь с внешними фазами ЖЦ. Назначение и содержание 12 архитектурных блоков. Критерии выбора конкретного метода описания. Понятие реинжиниринга.

Функциональный ракурс. Понятие функции. Элементарные функции и дерево функций. Объектно-ориентированный, процессно-ориентированный и операционно-ориентированный подходы к составлению дерева функций. Y-диаграмма, Диаграмма целей. Диаграммы уровня спецификации требований и описания реализации.

Организационный ракурс. Организационные диаграммы. Диаграммы распорядка дня. Управленческий/процессный ракурс. Диаграмма EPC. Событийно-ориентированный подход к описанию бизнес-процессов. Правила ветвления бизнес–процессов.

Диаграммы ARIS для составления BSC.

Модуль 6. Основы теории принятия решений

Человек в контуре организационного управления. Понятие "человеческого фактора" в работе и управлении АС. Проблема принятия решений. Основные черты ситуаций, в которых требуется принятие решений.

Виды решений: формальные и творческие. Процесс принятия решения. Классификация задач принятия решений. Непараметрические, однокритериальные и многокритериальные ЗПР. Принятие решений в условиях определенности, риска и неопределенности. Общая постановка однокритериальной детерминированной ЗПР.

Проблема большого количество критериев для принятия решения. Интегральные критерии, ключевые показатели эффективности (KPI) и правила их построения.

BSC — Система сбалансированных показателей. Четыре основных аспекта (ракурса) ССП и взаимосвязь между ними. Оптимальное количество целевых показателей ССП по Каплану&Нортону и по Шнейдерману. Причины введения дополнительных аспектов. Аспект "поддержки" для некоммерческих организаций.

Балансировка карты показателей. Установка каузальных связей между целевыми показателями и критическими факторами. Логическая балансировка. Локальная оптимизация. Корректировка ССП во время эксплуатации.

Определение зависимостей стратегических целей от ключевых показателей эффективности. Иерархичность и каскадирование BSC. Ограничения BSC, трудность определения значений показателей ракурсов. Особенности применения BSC в России.

Модуль 7. Введение в стандарт ITIL

Назначение и область применения ITIL. Структура стандарта и назначение основных книг. ИТ-сервисы и бизнес-процессы. Соглашение об уровне предоставления сервиса (SLA). Пример управления сервисами.

Книга Service Delivery (Предоставление сервиса). Цели, метрики и деятельности процессов: Service Level Management (Управление уровнем сервиса), Capacity Management (Управление емкостями, возможностями), Continuity Management (Управление непрерывностью, рисками), Cost Management (Управление затратами), Availability Management (Управление доступностью).

Книга Service Support (Поддержка сервиса). Цели, метрики и деятельности процессов: Service Desk (Доска сервисов), Incident Management (Управление инцидентами), Problem Management (Управление проблемами), Configuration Management (Управление конфигурациями), Software Control & Distribution (Распространение и управление программным обеспечением), Change Management (Управление изменениями), Release Management (Управление релизами, версиями).

Назначение и структура книг Business Perspective (Перспектива бизнеса), ICT Infrastructure Management (Управление инфраструктурой ИКТ), Application Management (Управление приложениями).

Модуль 8. Введение в стандарт CobiT

Назначение и область применения CobiT. Контрольные объекты, информация и ИТ в трактовке CobiT. Структура стандарта и назначение основных книг. Процессный подход к управлению. ИТ ресурсы и информационные критерии.

Модель ЖЦ CobiT. Процессы, входящие в домены: планирование и организация, проектирование и внедрение, эксплуатация и сопровождение, мониторинг.

Принципы управления. Основные вопросы, решаемые CobiT на стратегическом и тактическом уровнях. Модели зрелости процессов. Шкала моделей и 6 уровней зрелости ИТ-процессов.

Соотношение бизнес целей и ИТ. Критические факторы успеха, ключевые индикаторы целей и результатов. Взаимосвязь ключевых показателей эффективности, BSC и инструментов CobiT.

Принципы аудита. Аудит ИТ как обратная связь контура управления. Внешний аудит. Этический кодекс аудитора. Структура принципов аудита. Разделение на уровни.

Взаимосвязь CobiT и других стандартов (ITIL, RUP, ГОСТы 34, 19 и ИСО'МЭК, 9000 серий и др.).

Правила расчета экзаменационных оценок

Вопросы к экзамену разделены на три уровня сложности. Примерное соотношение знание разделов каждого модуля и максимальной оценки приведено ниже в таблице.

Оценка	***Модуль***	***Темы, разделы***
Удовлетворительно	*1, 2*	*все*
	3, 4	*1*
	5	*1, 3*
	6	*1–2*
	7, 8	*1*
Хорошо	*1,2*	*все*
	3	*1–4*
	4	*1–2*
	5	*1–3*
	6	*1–4*
	7	*1–3*
	8	*1–5*
Отлично	*1–8*	*все*

Указания по выполнению домашнего задания

Цель задания

Приобрести практические навыки работы с моделями ARIS (ARchitecture of Informftion Systems) на примере организационных диаграмм в рамках Organization View, освоить методы работы с деловой графикой в пакете MS Visio (или Aris EasyDesign), изучить и применить на практике механизмы поиска и систематизации информации в Интернет.

Порядок выполнения

- Изучить теоретическое введение (лекции и рекомендуемую литературу).
- Последовательно выполнить все пункты домашнего задания.
- Оформить отчет по домашнему заданию.

Задание

Выбрать из предложенного списка три вуза (институт, академию и университет). Проверить наличие в Интернет необходимой для выполнения информации и согласовать выбранные варианты со старостой (или другим ответственным лицом группы).

Разработать условно-графические обозначения основных блоков моделей Aris в Visio. (Рекомендуется блоки сохранить в виде Stencils). При использовании программных средств Aris этот этап может быть опущен.

Собрать всю необходимую информацию через Интернет и сохранить в папке "Исходные данные", разделив ее на три папки по каждому вузу.

Разработать организационные модели (укрупненную, представляющую структуру вуза, и детализированные по отдельным подразделениям с указанием должностей, ролей и другой информации). Записать разработанные модели в формате MS Visio в папку "Модели".

Разработать отчет в формате DOC и HTML. Распечатать отчет и записать его на компакт диск в папку с уникальным названием следующего вида: ИУ5_НомерГруппы_ФамилияИО. Допускается подготовить коллективный CD-ROM с несколькими заданиями.

Замечание! Разрешается заменить один из вузов на другую организацию с численностью более 100 человек.

Содержание отчета

- Титульный лист в соответствии с ГОСТ ЕСПД (документ пояснительная записка или техническое задание).
- Название, цель работы и задания.
- Краткое описание предметной области и выбранной задачи.
- Краткие данные об источнике информации: адресы Интернет, количество страниц, оценка полноты и т.д.
- Ключевые исходные данные.
- Разработанные модели.
- Список сокращений.
- Литература.

Указания по выполнению курсовой работы

Цель работы

Приобрести практические навыки работы с методами и стандартами по информационным технологиям в областях, охватывающих жизненный цикл автоматизированных систем и программного обеспечения, а также освоить основные классические математические методы теории принятия решений.

Порядок выполнения работы

- Изучить теоретическое введение (лекции и рекомендуемую литературу).
- Последовательно выполнить все пункты курсовой работы.
- Оформить отчет по курсовой работе.

Задания

Выбрать в произвольной предметной области задачу консалтинга, создания, внедрения, сопровождения или другого процесса, связанного с разработкой программного средства, автоматизированной системы или ИТ-услуги, которая позволяет выполнить все задания курсовой работы. Дать краткое естественно-языковое описание предметной области и предлагаемой задачи. Рекомендуется выбрать задачу, связанную с разработкой web-сайта.

Формализовать описание предметной области с использованием моделей ARIS.

Разработать организационную структуру предприятия, для которой решается описанная задача (разработка сайта).

Определить требования к решаемой задаче (к функциям, структуре, составу сайта и т.д.).

Описать не менее 3-х ключевых бизнес-процессов решаемых задач (с использованием сайта или без него).

Выделить задачу принятия решения (например, по выбору технических или программных средств для создания и эксплуатации сайта или оказания соответствующих ИТ-услуг). Выделить (разработать) не менее 5 критериев и предложить не менее 5 альтернативных вариантов задачи принятия решения. Провести анализ наиболее предпочтительных вариантов решений с использованием всех рассмотренных на лекциях методов классической Теории принятия решений (ТПР).

Усложнить задачу принятия решения, разработав 8–12 целевых критериев (целей и критических факторов) для 4–5 стандартных ракурсов BSC. Построить каузальные отношения между целевыми критериями и

описать правила их определения с помощью двух-трех KPI. Для описания BSC рекомендуется использовать ARIS.

Спроецировать BSC на организационную структуру предприятия и произвести каскадирование по уровням управления.

Разработать ТЗ на АС или ПО в соответствии с ГОСТами серий 34, 19 или ИСО'МЭК.

Разработать Соглашение о представлении сервисов (SLA), включающее описание не менее 3-х ИТ-услуг (например, по разработке, сопровождению, модернизации сайта) на базе рекомендаций ITIL.

Разработать отчет в формате DOC и HTML. Распечатать отчет и записать его на компакт диск в папку с уникальным названием следующего вида: ИУ5_НомерГруппы_ФамилияИО. Допускается подготовить коллективный CD-ROM с несколькими заданиями.

Содержание отчета

- Титульный лист в соответствии с ГОСТ ЕСПД (документ пояснительная записка или техническое задание).
- Название, цель работы и задания.
- Пункты задания.
- Список сокращений.
- Литература.

Часто задаваемые вопросы (FAQS)

Q *В чем разница между стандартами "Де-факто" и "Де-юре"?*

A Стандарты "Де-юре" — это нормативные документы, утвержденные государством или международными организациями стандартизации. К ним, например, относятся ГОСТы.

Стандарты "Де-факто" — это те стандарты, которые реально (широко) используются. Например, ITIL, COBIT.

Q *Как лучше готовиться к вопросам по ГОСТам (модули 3 и 4)?*

A Чтобы лучше запомнить и понять ГОСТы, необходимо попытаться приложить каждый пункт их содержания к уже известным вам задачам по созданию ПО и АС. Хорошим помощником могут оказаться ваши курсовые работы. При таком подходе могут сформироваться устойчивые ассоциации и более глубокое понимание ГОСТов.

Q *Какие типичные ошибки возникают при выполнении курсовой работы?*

A 1. При моделировании иерархической организационной структуры предприятия часто вместо специальных элементов «организационная единица» используют элементы модели «группа» и «персонал», что не допускается синтаксисом и блокируется в системе моделирования Aris на программном уровне. При построении моделей в Visio такая проверка не осуществляется, и многие привязывают к персоналу (должности) или группе подчиненные подразделения.

2. При описании бизнес-процессов не указывается, кто выполняет действия, как документы и другие артефакты передаются и где используются. Во избежание этих ошибок нужно стремиться, чтобы бизнес-процессы начинались с событий, а каждое действие (бизнес-функция) было привязано к исполнителю и связано с выходными и входными данными (документами, артефактами).

3. При решении задачи с использованием методов ТПР часто неправильно определяют области компромисса и согласия.

4. При составлении карт BSC многие используют ракурс «внешние процессы», что неправильно, т.к. если они имеют отношение к деятельности организации, то они не могут считаться внешними, а в противном случае они не должны рассматриваться вовсе.

5. При оформлении ТЗ часто возникает путаница с неправильной разбивкой требований к техническим и программным средствам, не указывается информация о выходной документации (не определяется минимальный комплект) и сроках (периодах) выполнения работ.

6. Соглашение об уровне представления сервиса (SLA) не направлено на выполнение рекомендаций и требований ITIL, мало связано с ИТ и чаще всего напоминает финансовый документ (договор), правила написания которого в рамках курса не изучались.

Q *Какие дополнительные пункты могут присутствовать в работе?*

A
- Использование стандартов и рекомендаций CobiT.
- Составление плана проекта в соответствии с требованиями RUP или ГОСТ 34.
- Реализация плана выполнения проекта в среде MS Project.
- Оформление дополнительной документации по разрабатываемому программному средству согласно требованиям ГОСТ.
- Рассмотрение стандартов качества ISO 9000 при описании бизнес-процессов.

Q *Можно ли в качестве предприятия, для которого решается задача, выбрать абстрактную, несуществующую организацию?*

A Можно выбрать несуществующее предприятие, но при описании

должна соблюдаться логика в его деятельности, структуре и бизнес-процессах.

Q *В ГОСТ на техническое задание говорится, что "В разделе «Технико-экономические показатели» должны быть указаны: ориентировочная экономическая эффективность...". Если основная тема курсовой работы — разработка сайта, то можно ли экономическую эффективность как-то привязать к продажам, идущим через сайт и т.п. Что под ней подразумевается в других случаях?*

A Можно привязать экономическую эффективность к продажам, если создание сайта приведет к их росту. В общем случае технико-экономические показатели берутся из документа технико-экономическое обоснование и определяют экономический (и технический) эффект от внедрения П/О.

Если такого эффекта нет, то целесообразность автоматизации отсутствует. В реальности определить эффективность и выделить показатели достаточно трудная задача. Типовые показатели можно встретить в литературе: ROI — return on investment (возврат инвестиций), совокупная стоимость владения и т.д.

Q *Нужно ли указывать имена и фамилии конкретных сотрудников в организационной структуре предприятия и бизнес-процессах или можно ограничиться должностями?*

A Можно ограничиться должностями.

Q *Если курсовая работа посвящена разработке программного средства для предприятия, то обязательно ли все три бизнес-процесса должны относиться к этому программному средству или 1–2 могут описывать другие аспекты деятельности предприятия (например, процедуру подготовки вводных данных для программного средства)?*

A Не обязательно, но желательно.

Q *Согласно какому документу следует оформлять титульный лист SLA?*

A Особые требования к титульному листу SLA не предъявляются.

Q *При составлении BSC нужно подобрать 2–3 KPI для каждой цели и критического фактора, или же их всего должно быть 2–3 для всех, а отличия будут только в расстановке коэффициентов?*

A Придумывать множество (10-12) целей и факторов на основе 2–3 показателей нецелесообразно, поэтому для каждой цели должны существовать свои показатели (KPI), но они могут пересекаться.

Q *Что такое критический фактор?*

A Критический фактор (в рамках системы сбалансированных пока-

зателей) — это нецелевой показатель, который необходимо учитывать в стратегии. Он может быть негативным или положительным. Например, рост инфляции, рост уровня информатизации на предприятиях клиентов, увеличение среднего возраста сотрудников.

Q *Можно ли проектировать BSC и другие схемы в Visio?*

A Да, это допускается, однако Visio не имеет встроенных средств проверки корректности моделей.

Q *Что понимается под "Проблемой большого количества критериев для принятия решения" (модуль 6 вопрос 3)?*

A Реальные системы описываются большим количеством показателей, которое может превосходить несколько тысяч единиц. Принять решение по такому количеству показателей крайне трудно, поэтому и возникает проблема. Для ее решения используют многие способы, некоторые из которых рассматриваются в курсе.

Q *Обязательно ли описывать, как выбрано то или иное значение оценки для критерия, если это в принципе очевидно?*

A Желательно указать источник данных или принцип определения оценок.

Q *Чем вектор приоритета отличается от весового вектора?*

A В рамках материала, рассмотренного на лекциях, вектор приоритета показывает относительную важность локального критерия по сравнению с рядом стоящими критериями. Весовой вектор в общем случае не привязан к расположению.

Q *Можно ли использовать нормировку оценок без указания реальных данных? Например, если в качестве критерия выступает прибыль, то не писать ее реальное значение и потом нормировать, а оценивать прибыль сразу в процентах. Можно ли использовать абстрактные цифры (данные)?*

A Можно использовать абстрактные данные, но они должны быть логичными и непротиворечивыми.

Q *Можно ли рассматривать однокритериальную и многокритериальную ЗПР на одной и той же матрице оценок?*

A Обычно значениям критериев сопоставляются оценки, а не наоборот. В многокритериальной задаче, безусловно, можно включить значения критерия из однокритериальной ЗПР, но значения остальных критериев должны быть новыми.

Q *Какие книги по теории принятия решений вы можете рекомендовать?*

A Для изучения классической ТПР можно выбрать любой рекомендованный УМО учебник или учебное пособие. Из последних книг,

которые интересны также другими аспектами и связью с ИТ, можно порекомендовать следующие:

Хемди А. Таха. Введение в исследование операций. 7-е изд. 2005, Вильямс. 912 стр. (Для изучения основ можно обратиться к более раннему изданию, которое есть в любой технической библиотеке).

Ларичев О.И. Теория и методы принятия решений а также Хроника событий в Волшебных Странах. М.: «Логос», 2000, 296 с., с илл.

Из старых книг можно также посмотреть:

Макаров И.М., Виноградская Т.М., Рубчинский А.А., Соколов В.Б. Теория выбора и принятия решения, М.: Наука, 1982. 328 с.

Q *Какие периодические издания, отражают современные аспекты в области АСОИУ.*

A Журналы: "Открытые системы", "Cnews", "Intelligence Interprise", "Byte","PC Week" и др.

Q *В какой литературе (кроме лекций) можно найти ответы на вопросы о том, что такое архитектура АСОИУ, архитектура как каркас и совокупность принципов построения АС, архитектура как искусство проектирования и создания информационных и автоматизированных систем?*

A Начать изучение этих вопросов надо с толкового или энциклопедического словаря, где представлены толкования и определения основных понятий. После четкого понимания, что такое система, АС, обработка информации, ИТ, управление, АСОИУ, архитектура, совокупность, каркас, искусство и т.д., необходимо обратиться к ГОС-Там, где даются специализированные определения.

Основные принципы построения АС можно найти в книгах *Менькова А.В., Четверикова В.Н., Мамиконова А.Г.* Современные подходы к ПО, ИС и АС можно найти в стандартах (серии ИСО-МЭК, ITIL, COBIT) и методологиях (ARIS, RUP, Oracle METHOD, ASAP и др.).

Рекомендую также книги, представленные на Интернет-странице http://www.intuit.ru/department/se/devis/lit.html:

Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем \\ Интернет-университет информационных технологий - ИНТУИТ.ру, 2005.

Данилин А., Слюсаренко А. Архитектура и стратегия. "Инь" и "янь" информационных технологий \\ Интернет-университет информационных технологий - ИНТУИТ.ру, 2005.

Список ГОСТ

Комплекс «новых» стандартов и руководящих документов на автоматизированные системы.

ГОСТ 34.003-90. Автоматизированные системы. Термины и определения. Automated systems. Terms and definitions.

ГОСТ 34.201-89 Виды, комплектность и обозначение документов при создании автоматизированных систем. Types, sets, and indication of documents for automated systems making.

ГОСТ 34.401-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Средства технические периферийные автоматизированных систем дорожного движения. Типы и технические требования.

ГОСТ 34.601-90 Автоматизированные системы. Стадии создания. Set of standards for automated systems. Stages of development.

ГОСТ 34.602-89 Общие положения, состав, содержание и правила оформления документа "Техническое задание на создание системы". Technical directions for automated system making.

ГОСТ 34.603-92 Информационная технология. Виды испытаний автоматизированных систем. Information technology. Types tests automated systems.

РД 50-34.698-90 Автоматизированные системы. Требования к содержанию документов.

Комплекс «старых» стандартов и руководящих документов на автоматизированные системы.

ГОСТ 24.104-85 Единая система стандартов автоматизированных систем управления. Автоматизированные системы управления. Общие требования. – Взамен ГОСТ 17195-76, ГОСТ 20912-75. ГОСТ 24205-80 (в части разд. 3 заменен ГОСТ 34.603-92).

ГОСТ 24.301-80 Система технической документации на АСУ. Общие требования к текстовым документам.

ГОСТ 24.302-80 Система технической документации на АСУ. Общие требования к выполнению схем.

ГОСТ 24.303-80 Система технической документации на АСУ. Обозначения условные графические технических средств.

ГОСТ 24.304-82 Система технической документации на АСУ. Требования к выполнению чертежей.

ГОСТ 24.401-80 Система технической документации на АСУ. Внесение изменений.

ГОСТ 24.402-80 Система технической документации на АСУ. Учет, хранение и обращение.

ГОСТ 24.501-82 Автоматизированные системы управления дорожным движением. Общие требования.

ГОСТ 24.701-86 Единая система стандартов автоматизированных систем управления. Надежность автоматизированных систем управления. Основные положения. – Взамен ГОСТ 24.701-83.

ГОСТ 24.702-85 Единая система стандартов автоматизированных систем управления. Эффективность АСУ. Основные положения.

ГОСТ 24.703-85 Единая система стандартов автоматизированных систем управления. Типовые проекты решения в АСУ. Основные положения.

Единая система Конструкторской документации (ЕСКД)

ГОСТ 2.106-96 Текстовые документы. Textual documents.

Единая система программной документации (ЕСПД)

ГОСТ 19.001-77 Единая система программной документации. Общие положения.

ГОСТ 19.005-85 Единая система программной документации. Р-схемы алгоритмов и программ. Обозначения условные графические и правила выполнения.

ГОСТ 19.101-77 (СТ СЭВ 1626-79) Единая система программной документации. Виды программ и программных документов

ГОСТ 19.102-77 Единая система программной документации. Стадии разработки.

ГОСТ 19.103-77 Единая система программной документации. Обозначения программ и программных документов.

ГОСТ 19.104-78 (СТ СЭВ 2088-80) Единая система программной документации. Основные надписи.

ГОСТ 19.105-78 (СТ СЭВ 2088-80) Единая система программной документации. Общие требования к программным документам.

ГОСТ 19.106-78 (СТ СЭВ 2088-80) Единая система программной документации. Требования к программным документам, выполненным печатным способом.

ГОСТ 19.201-78 (СТ СЭВ 1627-79) Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

ГОСТ 19.202-78 (СТ СЭВ 2090-80) Единая система программной документации. Спецификация. Требования к содержанию и оформлению.

ГОСТ 19.301-79 (СТ СЭВ 3747-82) Единая система программной документации. Программа и методика испытаний. Требования к содержанию и оформлению.

ГОСТ 19.401-78 (СТ СЭВ 3746-82) Единая система программной документации. Текст программы. Требования к содержанию и оформлению.

ГОСТ 19.402-78 (СТ СЭВ 2092-80) Единая система программной документации. Описание программы.

ГОСТ 19.403-79 Единая система программной документации. Ведомость держателей подлинников.

ГОСТ 19.404-79 Единая система программной документации. Пояснительная записка. Требования к содержанию и оформлению.

ГОСТ 19.501-78 Единая система программной документации. Формуляр. Требования к содержанию и оформлению.

ГОСТ 19.502-78 (СТ СЭВ 2093-80) Единая система программной документации. Описание применения. Требования к содержанию и оформлению.

ГОСТ 19.503-79 (СТ СЭВ 2094-80) Единая система программной документации. Руководство системного программиста. Требования к содержанию и оформлению

ГОСТ 19.504-79 (СТ СЭВ 2095-80) Единая система программной документации. Руководство программиста. Требования к содержанию и оформлению.

ГОСТ 19.505-79 (СТ СЭВ 2096-80) Единая система программной документации. Руководство оператора. Требования к содержанию и оформлению.

ГОСТ 19.506-79 (СТ СЭВ 2097-80) Единая система программной документации. Описание языка. Требования к содержанию и оформлению.

ГОСТ 19.507-79 (СТ СЭВ 2091-80) Единая система программной документации. Ведомость эксплуатационных документов.

ГОСТ 19.508-79 Единая система программной документации. Руководство по техническому обслуживанию. Требования к содержанию и оформлению.

ГОСТ 19.601-78 Единая система программной документации. Общие правила дублирования, учета и хранения.

ГОСТ 19.602-78 Единая система программной документации. Правила дублирования, учета и хранения программных документов, выполненных печатным способом.

ГОСТ 19.603-78 (СТ СЭВ 2089-80) Единая система программной документации. Общие правила внесения изменений.

ГОСТ 19.604-78 (СТ СЭВ 2089-80) Единая система программной документации. Правила внесения изменений в программные документы, выполненные печатным способом.

ГОСТ 19.701-90 (ИСО 5807-85) Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения - Взамен ГОСТ 19.002-80, ГОСТ 19.003-80.

Комплекс стандартов на разработку программного обеспечения (аналоги стандартов ISO)

ГОСТ Р ИСО'МЭК 12119-2000 - Информационная технология. Пакеты программ. Требования к качеству и тестирование.

ГОСТ Р ИСО'МЭК 12207-99 - Информационная технология. Процессы жизненного цикла программных средств.

ГОСТ Р ИСО'МЭК ТО 9294-93 - Информационная технология. Руководство по управлению документированием программного обеспечения.

ГОСТ Р ИСО'МЭК 15910-2002 - Информационная технология. Процесс создания документации пользователя программного средства.

ГОСТ Р ИСО'МЭК ТО 12182-2002 - Информационная технология. Классификация программных средств.

ГОСТ Р ИСО'МЭК 14764-2002 - Информационная технология. Сопровождение программных средств.

ГОСТ Р ИСО'МЭК 15026-2002 - Информационная технология. Уровни целостности систем и программных средств.

ГОСТ Р ИСО'МЭК ТО 15271-2002 - Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207 (Процессы жизненного цикла программных средств).

С.А.Кесель

ИСПОЛЬЗОВАНИЕ АППАРАТНЫХ МОДУЛЕЙ БЕЗОПАСНОСТИ ДЛЯ ЗАЩИТЫ ИНФОРМАЦИИ

Методические указания
к лабораторной работе по дисциплине
«Защита информации в АСОИУ»

Цель работы.

Ознакомиться с принципами работы и характеристиками аппаратных модулей безопасности и приобрести практические навыки по работе с криптографическими ключами.

Порядок выполнения работы.

1. Изучить теоретическую часть.
2. Последовательно выполнить все задания к лабораторной работе.
3. Проверить правильность выполнения заданий.
4. Оформить отчет по лабораторной работе.

Задания к лабораторной работе

1. Ознакомиться с техническими характеристиками модулей безопасности *protectserver orange, protectserver gold, protectserver external* компании Eracom (SafeNet).

2. Ознакомиться на официальном сайте NIST (Американского института стандартов и технологий) с сертификатами и политиками безопасности рассматриваемых модулей. http://csrc.nist.gov/cryptval/140-1/1401vend.htm

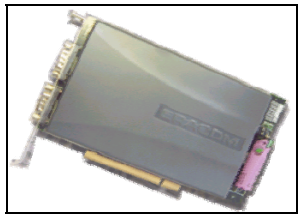
3. Запустите программный эмулятор работы модуля безопасности ***protectserver orange*** и осуществить генерацию и экспорт криптографических ключей согласно индивидуальному заданию.

Содержание отчета.

1. Название и цель работы.
2. Задание.
3. Краткое описание модулей безопасности и основных операций по управлению криптографическими ключами.
4. Протоколы проведенных операций по работе с ключами.

Теоретическая часть

protectserver orange



- ***protectserver orange*** компании ***eracom*** представляет собой новое поколение интеллектуальных криптографических адаптеров на основе PCI шины, которые используются для криптографической обработки данных и хранения ключей.
- ***protectserver orange*** сочетает в себе характеристики аппаратного модуля безопасности (Hardware Security Module) с преимуществами, производительностью и скоростью работы аппаратного крипто акселератора (Hardware Crypto Accelerator).
- ***protectserver orange*** базируется на базовой архитектуре безопасности ***protectcore orange.***

Сфера применения.

Компоненты PKI такие как CAs, RAs, директории, Time Stamping серверы и OCSP респондеры.

Системы электронной коммерции, торговые системы и Payment Gateways.

Интернет провайдеры и провайдеры сервисных приложений.

SSL крипто акселератор для обеспечения безопасности вэб серверов.

Firewalls и VPN.

Системы для безопасного/конфиденциального хранения документов и документооборота.

Удостоверяющие серверы и серверы для хранения ключей.

Электронные подписи под документами и системы нотарификации

Аппаратный модуль безопасности (Hardware Security Module) для банкоматов для обеспечения безопасности розничных платежей.

Аппаратный модуль безопасности (Hardware Security Module) глобальной платежной системы Identrus.

Любые другие приложения, требующие безопасного обмена данными, хранения и управления ключами или надежной аудентификации.

Функциональные характеристики.

Алгоритмы, соответствующие FIPS и режими без FIPS.

Сертификаты публичных ключей (public key) — X.509 v3, запросы по сертификату PKCS#10.

Программируемые интерфейсы приложений (APIs) и стандартная совместимость с — PKCS#11, DSS, SHS, ANSI X.9, алгоритмами, содержащимися в open SSL/TLS/WTLS, IPSec, SET, S/MIME.

Совместимость со следующими приложениями — Identrus, Entrust, Sun, Netscape, ValiCert, Apache/OpenSSL, Microsoft IIS.

Расширенные возможности по управлению ключами and высокий уровень физической защиты при хранении ключей.

Поддержка ассиметричных ключей длиной до 4096 бит.

Часы реального времени и генератор случайных чисел находятся на плате внутри криптографического микропроцессора.

Безопасный экспорт/импорт ключей с помощью смарт карты.

Объем памяти для безопасного хранения ключей, дополнительно снабженной внешним гальваническим элементом, составляет 1 Мбайт.

Обеспечение безопасностного обновления соответствующего программного обеспечения адаптера самим потребителем на месте работы оборудования.

Платформы — Windows NT/2000, Sun Solaris, Linux, SCO OpenServer and UnixWare.

Поддерживает современные криптоалгоритмы:

Симмертичные —AES, DES, 3DES, IDEA, CAST-128, RC2, RC4 (для режимов ECB, CBC, stream (RC4))

Ассимметричные — *Для цифровой подписи* — RSA-raw, RSA-EMSA-PKCS, ISO-9796, DSA, ECDSA (ANSI X9.62); *для шифрования/дешифрования* — RSA-raw, RSA-EMS-OAEP, RSA-EME-PKCS, RSA-OAEP-SET (PKCS#1 v2.0), ECDH (ANSI X9.62).

Управление ключами Diffie-Hellman.

Аутентификация сообщений — DES-MAC (ANSI X9.9), DES-MAC (ANSI X9.19), 3DES-MAC, IDEA-MAC, HMAC-MD2, HMAC-MD5, RIPEMD128, HMAC-RIPEMD160, HMAC-SHA-1.

Функции хэширования — MD2, MD5, SHA-1, RIPEMD-128, RIPEMD-160.

Сертификаты открытых ключей соответствуют — PKCS #10 certificate requests, X.509 v3, PKCS #7 decode

Интерфейсы.

PCI шина и асинхронные сериальные порты.

Сертификация.

Продукт сертифицирован согласно стандарту FIPS 140-1 Level 3 и также обладает следующими сертификатами производителей и платежных систем: Valicert, Secude, Entrust Ready и Identrus Compliant.

Преимущества продукта PSO.

•XCM ***eracom*** 6-го поколения.

•Различные уровни производительности (до 450 RSA private key операций/секунду).

•Крипто акселерация.

•Поддерживает ключи RSA длиной до 4096 бит.

•Поддержка всех стандартных APIs.

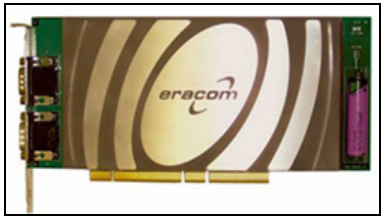
•Стандартная поддержка всех основных аппаратных и программных платформ.

•Возможность модернизации firmware на месте работы устройства самим потребителем.

•Сертификация в области информационной безопасности (FIPS 140-1 level 3, CC).

•Кастомизация / Расширение функциональных возможностей.

ProtectServer Gold.



ProtectServer Gold – это надежный аппаратный модуль безопасности, работающий на шине PCI, предназначенный для быстрой обработки криптографических процессов защиты в серверных системах с поддержкой приложений, требующих высокой производительности при выполнении симметричных и несимметричных криптографических операций. Он оснащен 64-разрядным интерфейсом PCI, защищенной памятью для хранения данных объемом 4 Мб и специальным криптографическим процессором, предназначенным для обработки криптографических операций и транзакций с высокой скоростью.

В модуле представлен широкий выбор служб, включая быстродействующее шифрование, аутентификацию пользователя и данных, сохранность сообщений, безопасное хранение ключей и управление ключами для систем электронной торговли, инфраструктур открытых ключей,

систем управления документами, систем электронного управления счетами EBPP, шифрования баз данных, финансовых транзакций EFT и многих других приложений.

Модуль HSM сохраняет все секретные ключи и конфиденциальные данные в защищенной памяти и выполняет все конфиденциальные криптографические операции внутри собственной сертифицированной защищенной среды. Предлагаемый клиентам уровень безопасности хранения и обработки данных недоступен для альтернативных программных решений, при этом степень защиты конфиденциальности и целостности информации отвечает ожиданиям клиента и требованиям безопасности промышленных организаций.

Простота использования и управления ключами достигается за счет интуитивно понятного графического интерфейса пользователя (GUI). Применение смарт-карт обеспечивает максимальную защиту и удобное управление безопасным созданием архивных копий, восстановлением и передачей шифровальных ключей. Обновление программного обеспечения можно выполнять на рабочем месте, избегая затрат на отправку устройства к месту обслуживания.

Доступен широкий выбор интерфейсов прикладного программирования API, с помощью которых можно настроить криптографическое приложение в соответствии с отраслевыми стандартами безопасности и платформенной средой. Сюда входит широкий выбор функциональных наборов PKCS#11, которые доступны на рынке, приложения Java JCA/JCE и Microsoft CryptoAPI, предоставляемые поставщиком, а также средства органичной интеграции с пакетом OpenSSL. Эти средства являются дополнением к наборам команд обработки электронных денежных переводов и платежей и модулю с удобными функциями пользовательской настройки приложений, работающих в HSM.

Модуль **ProtectServer Gold** совместим с **ProtectServer Orange** для обеспечения преемственности с последними технологиями и стандартами безопасности.

Соответствие стандарту FIPS 140–2 уровня 3 (В настоящее время проводится сертификация на соответствие всем трем уровням стандарта FIPS 140-2) Соответствие стандарту FIPS 140-2 уровня 3 означает, что ProtectServer Gold обладает возможностями обнаружения попыток физического взлома и логических атак и защиты от них, а также выполняет безопасные процессы криптографии, включая верное применение отдельных коммерческих важных и утвержденных криптографических алгоритмов.

Надежная физическая защита, включающая чувствительные к взлому схемы, гарантирует удаление всей информации, хранящейся в защищенной памяти, в случае попытки физического проникновения в устройство.

Защищенная память с питанием от аккумулятора для хранения ключей 4 Мб защищенной памяти с питанием от аккумулятора для безопасного хранения и поддержки целостности симметричных ключей шифра, асимметричных ключей, сертификатов и конфиденциальных данных. Аккумулятор обеспечивает резервное питание электронного блока, реагирующего на несанкционированное вскрытие, при отключении питания системы. Любая попытка вмешательства, включая удаление из модуля HSM аккумулятора или удаление HSM из слота шины PCI, немедленно активирует уничтожение ключей из памяти.

Генератор случайных чисел ProtectServer Gold позволяет генерировать случайные криптографические ключи, которые гарантируют требуемый уровень энтропии, необходимый для высших уровней защиты. Это достигается с помощью генератора случайных чисел, отвечающего требованиям стандарта ANSI X9.31 и сертифицированного в соответствии с требованиями FIPS 140.

Передача ключа на смарт-карту. Сохранение криптографических ключей на смарт-карте снижает человеческий фактор возникновения ошибки при вводе ключей для нескольких модулей HSM и значительно ускоряет процесс настройки пула HSM с одинаковыми ключами в производственной среде с высокой доступностью. Кроме того, резервные копии на смарт-картах помогают восстановить секретные ключи после стирания памяти в случае попытки взлома, при обновлении, обслуживании или передаче ключа с одного HSM на другой.

protectserver external

protectserver external компании *eracom* представляет отдельный перифирийный back end сервер, предназначенный для криптографической обработки данных

protectserver external состоит из криптографического адаптера *protectserver gold* (максимальное число адаптеров внутри 2) и промыш-

ленного персонального компьютера в обязательном внешнем массивном стальном корпусе

protectserver external базируется на базовой архитектуре безопасности *protectcore orange.*

Сфера применения

Компоненты PKI такие как CAs, RAs, директории, Time Stamping серверы и OCSP респондеры

Системы электронной коммерции, торговые системы и Payment Gateways

Интернет провайдеры и провайдеры сервисных приложений

SSL крипто акселератор для обеспечения безопасности вэб серверов

Firewalls и VPN

Системы для безопасного/конфиденциального хранения документов и документооборота

Удостоверяющие серверы и серверы для хранения ключей

Электронные подписи под документами и системы нотарификации

Аппаратный модуль безопасности (Hardware Security Module) для банкоматов для обеспечения безопасности розничных платежей

Аппаратный модуль безопасности (Hardware Security Module) глобальной платежной системы Identrus

Любые другие приложения, требующие безопасного обмена данными, хранения и управления ключами или надежной аудентификации

Функциональные характеристики

Криптографический адаптер ***protectserver gold***

Алгоритмы – алгоритмы, соответствующие FIPS и режими без FIPS

Сертификаты публичных ключей (public key) - X.509 v3, запросы по сертификату PKCS#10

Программируемые интерфейсы приложений (APIs) и стандартная совместимость с - PKCS#11, DSS, SHS, ANSI X.9, алгоритмами, содержащимися в open SSL/TLS/WTLS, IPSec, SET, S/MIME

Совместимость со следующими приложениями - Identrus, Entrust, Sun, Netscape, ValiCert, Apache/OpenSSL, Microsoft IIS

Расширенные возможности по управлению ключами and высокий уровень физической защиты при хранении ключей

Поддержка ассиметричных ключей длиной до 4096 бит

Часы реального времени и генератор случайных чисел находятся на плате внутри криптографического микропроцессора

Безопасный экспорт/импорт ключей с помощью смарт карты

Объем памяти для безопасного хранения ключей, дополнительно снабженной внешним гальваническим элементом, составляет 1 Мбайт

Обеспечение безопасностного обновления соответствующего программного обеспечения адаптера самим потребителем на месте работы оборудования

Платформы - Windows NT/2000, Sun Solaris, Linux, SCO OpenServer and UnixWare

Промышленный персональный компьютер (ПК) во внешнем корпусе

Жесткий диск с флэш памятью в размере 128 Мбайт

128 Мбайт RAM

Интел совместимый процессор 300 Мгц

2 RS-232 сериальных порта

Электрический соединитель с кабелем (АС)

10/100 Мбит/сек Base-TX сетевой интерфейс с RJ45 LAN соединителем

Интерфейсы

Тип соединения: TCP/IP через Ethernet

Сертификация

Продукт сертифицирован согласно стандарту FIPS 140-1 Level 3 и также обладает следующими сертификатами производителей и платежных систем: Valicert, Entegrity, Secude, Entrust Ready и Identrus Compliant

Рекомендуемая литература

1. *Б.Шнаер* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.:ТРИУМФ, 2003. — 816с.

2. Математические и компьютерные основы криптологии: Учеб. пособие / Ю.С.Харин и др. Минск:Новое Знание, 2003. — 382с.

3. *Ю.В.Романец и др.* Защита информации в компьютерных системах и сетях Изд. 2-е. М.: Радио и связь, 2001. — 376с.

Научно-образовательный кластер
«Computer Linguistics–Artificial Intelligence–Multimedia»
(НОК CLAIM)

Научное издание

Интеллектуальные технологии и системы
Сборник учебно-методических работ и статей аспирантов и студентов

Выпуск 9

Составитель: *Филиппович Юрий Николаевич*

Редактирование и корректура выполнена авторами статей

Формат 60 × 84/16. Бумага офсетная. Гарнитура «Таймс». Печать на ризографе.
Усл.п.л. 20.5. Тираж 100 экз.