

# Интеллектуальные системы и технологии

Лекция 8.  
Экспертные системы

- **Экспертная система (ЭС, *expert system*)** — компьютерная система, способная частично заменить специалиста-эксперта в разрешении проблемной ситуации (из Википедии).
- Основным назначением ЭС является разработка программных средств, которые при решении задач, трудных для человека, получают результаты, не уступающие по качеству и эффективности решения, решениям получаемым человеком-экспертом.
- Начали разрабатываться в конце 1960-х годов.
- В 1980-х годах сложилась технология и методология разработки экспертных систем

- ЭС используются для решения так называемых неформализованных задач, общим для которых является то, что:
  - задачи не могут быть заданы в числовой форме;
  - цели нельзя выразить в терминах точно определенной целевой функции;
  - не существует алгоритмического решения задачи;
  - если алгоритмическое решение есть, то его нельзя использовать из-за
  - ограниченности ресурсов (время, память).

# Первые ЭС

- DENDRAL – ЭС для распознавания структуры сложных органических молекул по результатам их спектрального анализа (считается первой в мире экспертной системой);
- MOLGEN – ЭС для выработке гипотез о структуре ДНК на основе экспериментов с ферментами;
- XCON – ЭС для конфигурирования (проектирования) вычислительных комплексов VAX 11 в корпорации DEC в соответствии с заказом покупателя;
- MYCIN – ЭС диагностики кишечных заболеваний;
- PUFF – ЭС диагностики легочных заболеваний;
- MACSYMA – ЭС для символьных преобразований алгебраических выражений;
- YES/MVS – ЭС для управления многозадачной операционной системой MVS больших ЭВМ корпорации IBM;
- DART – ЭС для диагностики больших НМД корпорации IBM;
- PROSPECTOR – ЭС для консультаций при поиске залежей полезных ископаемых;
- POMME – ЭС для выдачи рекомендаций по уходу за яблоневым садом;

# Области использования ЭС

Сельское хозяйство

Бизнес

Химия

Коммуникации

Компьютерные системы

Образование

Электроника

Инжиниринг

Управление информацией

Производство

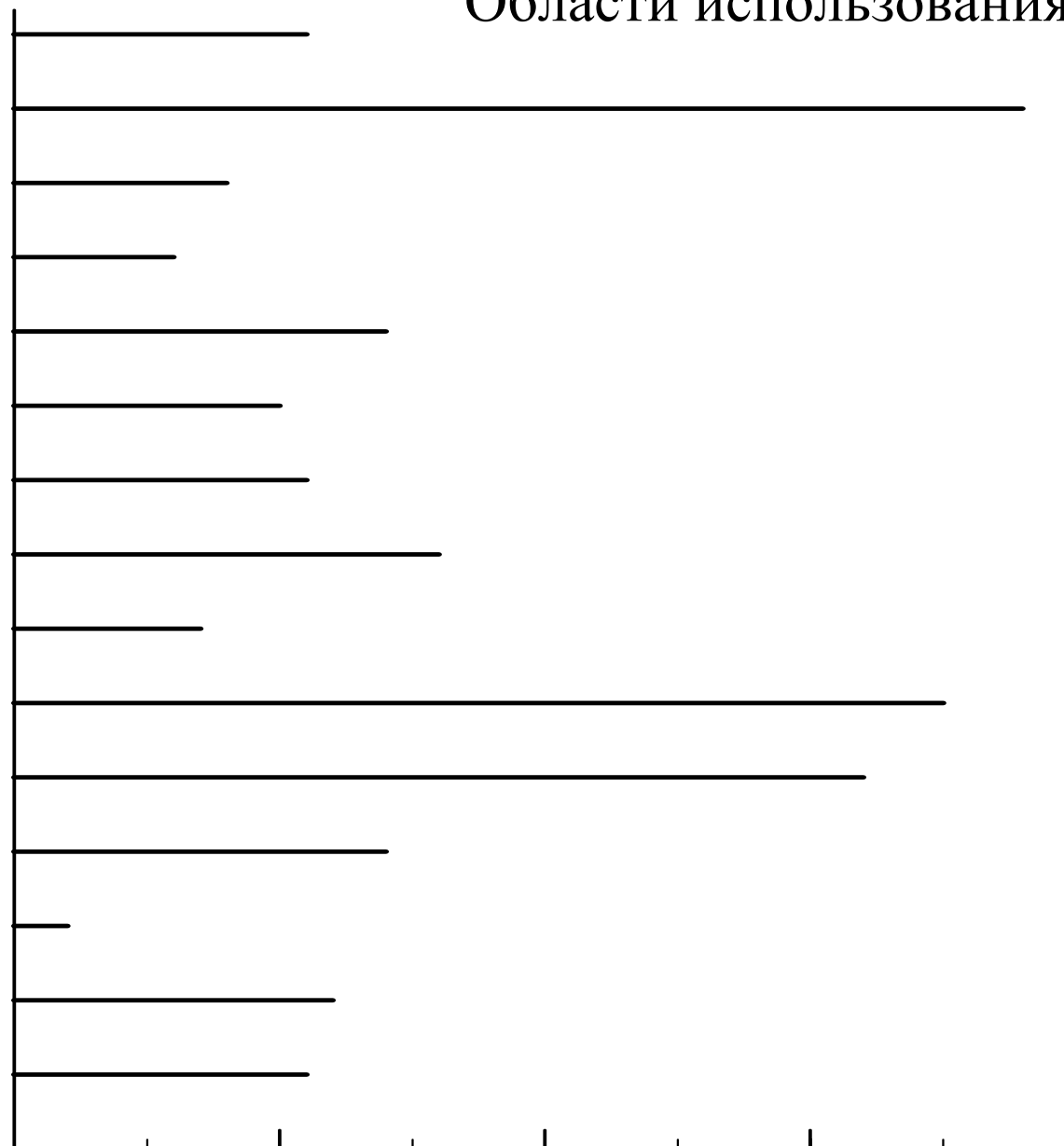
Медицина

Армия

Наука

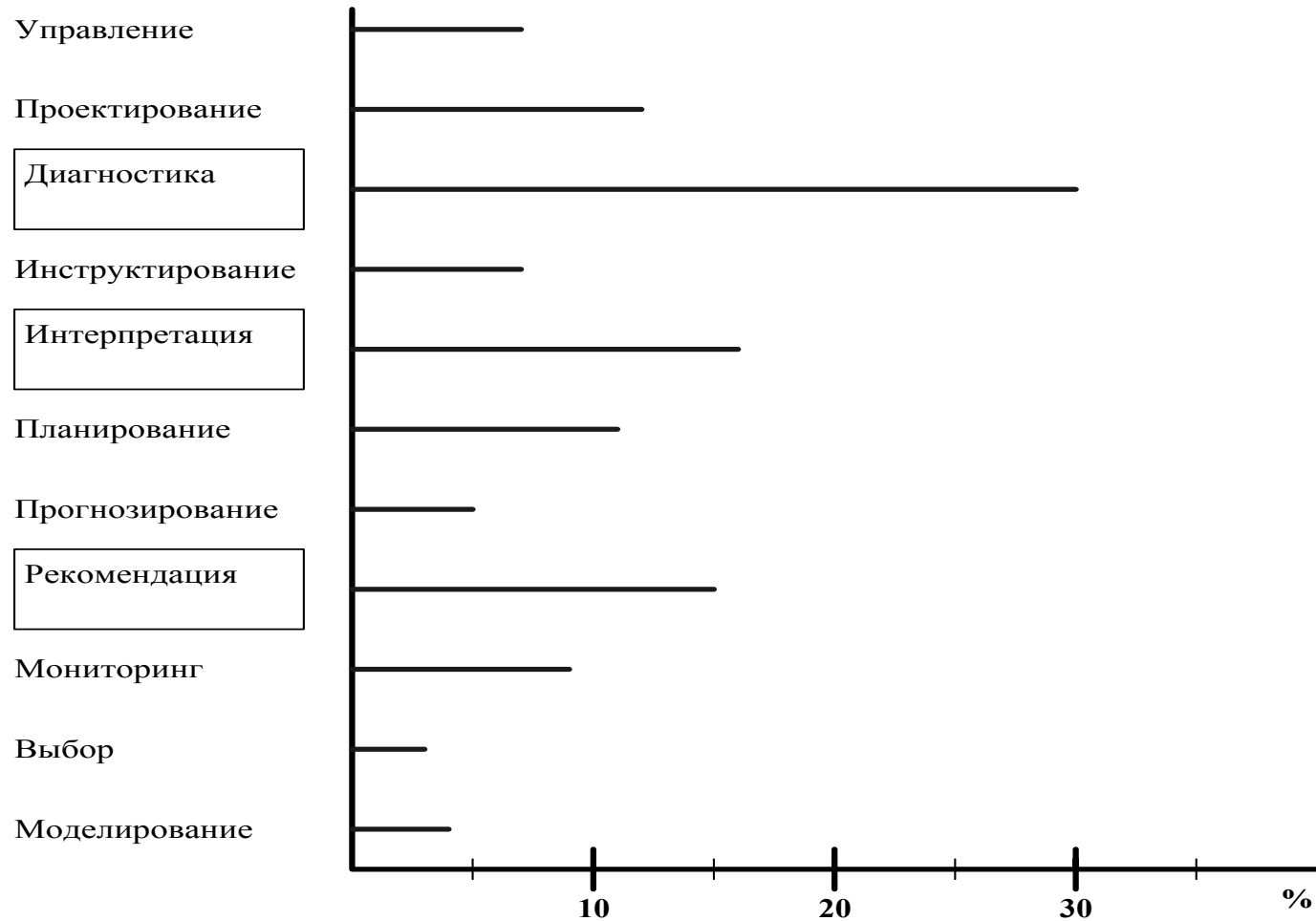
Космос

Транспорт



А.В.Гаврилов  
НГТУ, кафедра ВТ

# Задачи, решаемые ЭС



А.В.Гаврилов  
НГТУ, кафедра ВТ

# ЭКСПЕРТНЫЕ СИСТЕМЫ

ЗАДАЧА	ВРЕМЯ	КОМПЬЮТЕР	ИНТЕГРАЦИЯ
ДИАГНОСТИКА	СТАТИЧЕСКИЕ	СУПЕР ЭВМ	АВТОНОМНАЯ
ПРОЕКТИРОВАНИЕ	КВАЗИДИНАМИЧЕСКИЕ	СРЕДНЯЯ	ГИБРИДНАЯ
ПРОГНОЗИРОВАНИЕ	ДИНАМИЧЕСКИЕ	СИМВОЛЬНЫЙ ПРОЦЕССОР	
ПЛАНИРОВАНИЕ		РАБОЧАЯ СТАНЦИЯ	
ОБУЧЕНИЕ		ПЭВМ	

# Примеры экспертных систем

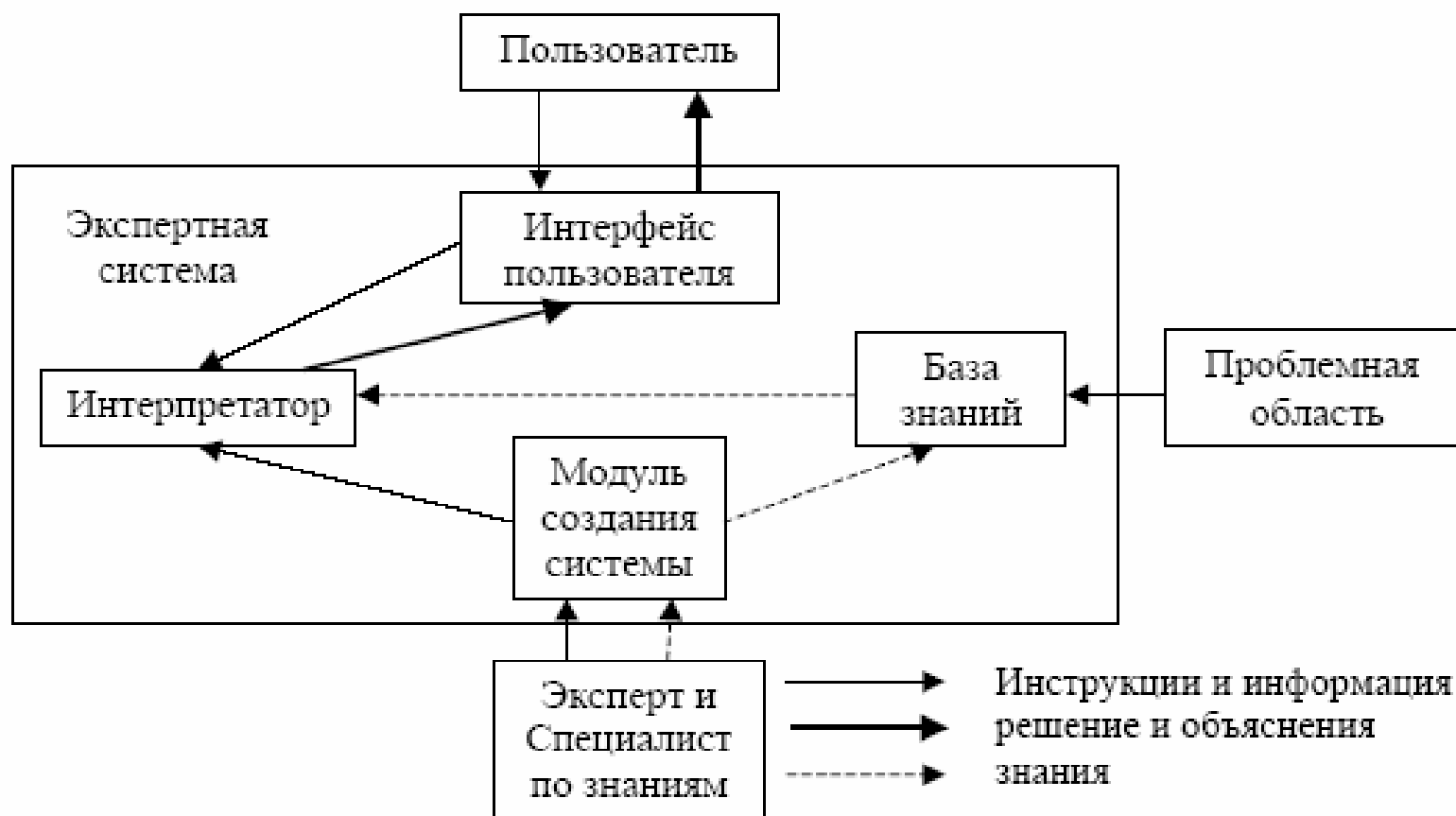
- **ACES.** Экспертная система выполняет картографические работы по нанесению обстановки на карты.
- **ASTA.** Экспертная система помогает аналитику определить тип радара, пославшего перехваченный сигнал.
- **DART.** Экспертная система помогает обрабатывать разведданные о центрах командования, управления и связи противника.
- **HANNIBAL.** Экспертная система выполняет оценивание ситуаций в области разведки радиообмена противника. Система идентифицирует соединения противника и боевой порядок их связи, интерпретируя данные радиоперехвата.
- **I&W.** Экспертная система помогает аналитикам из разведки предсказывать, когда и где произойдет следующее вооруженное столкновение.
- **CODES.** Экспертная система помогает разработчику базы данных, желающему использовать подход IDEF1 для определения концептуальной схемы базы данных.
- **MIXER.** Экспертная система оказывает помощь программистам в написании микропрограмм для разработанной Texas Instruments СБИС TI990. По заданному описанию микропрограммы система получает оптимизированные микропрограммы для TI990.
- **ACE.** Экспертная система определяет неисправности в телефонной сети и дает рекомендации по необходимому ремонту и восстановительным мероприятиям.



# Примеры экспертных систем (2)

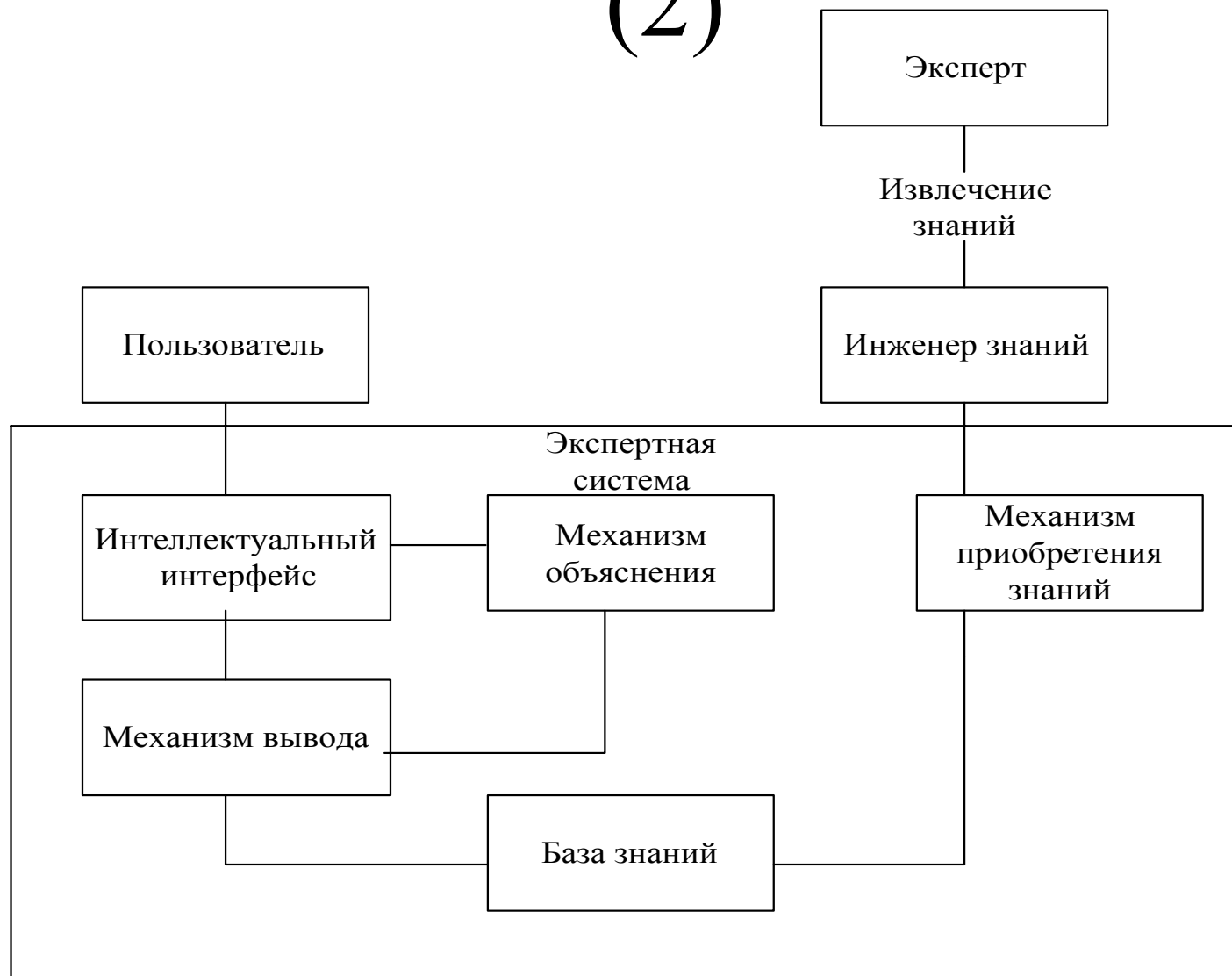
- S&PCBRS - разработана для решения следующих задач: оценка рейтинга ценных бумаг по данным о фирмах эмитентах; формирование корректной рейтинговой шкалы.
- Nereid - разработана для поддержки принятия решений для оптимизации работы с валютными опционами.
- ExpSystem PC 1.4.2 - поможет людям не столь хорошо разбирающимся в комплектующих для персонального компьютера подобрать себе приемлемую конфигурацию будущего компьютера
- FALCON - определяет вероятные причины отклонений параметров процесса от нормы на химическом заводе, интерпретируя данные, состоящие из числовых значений показаний приборов, положения переключателей и состояния аварийных датчиков

# Структура экспертной системы.



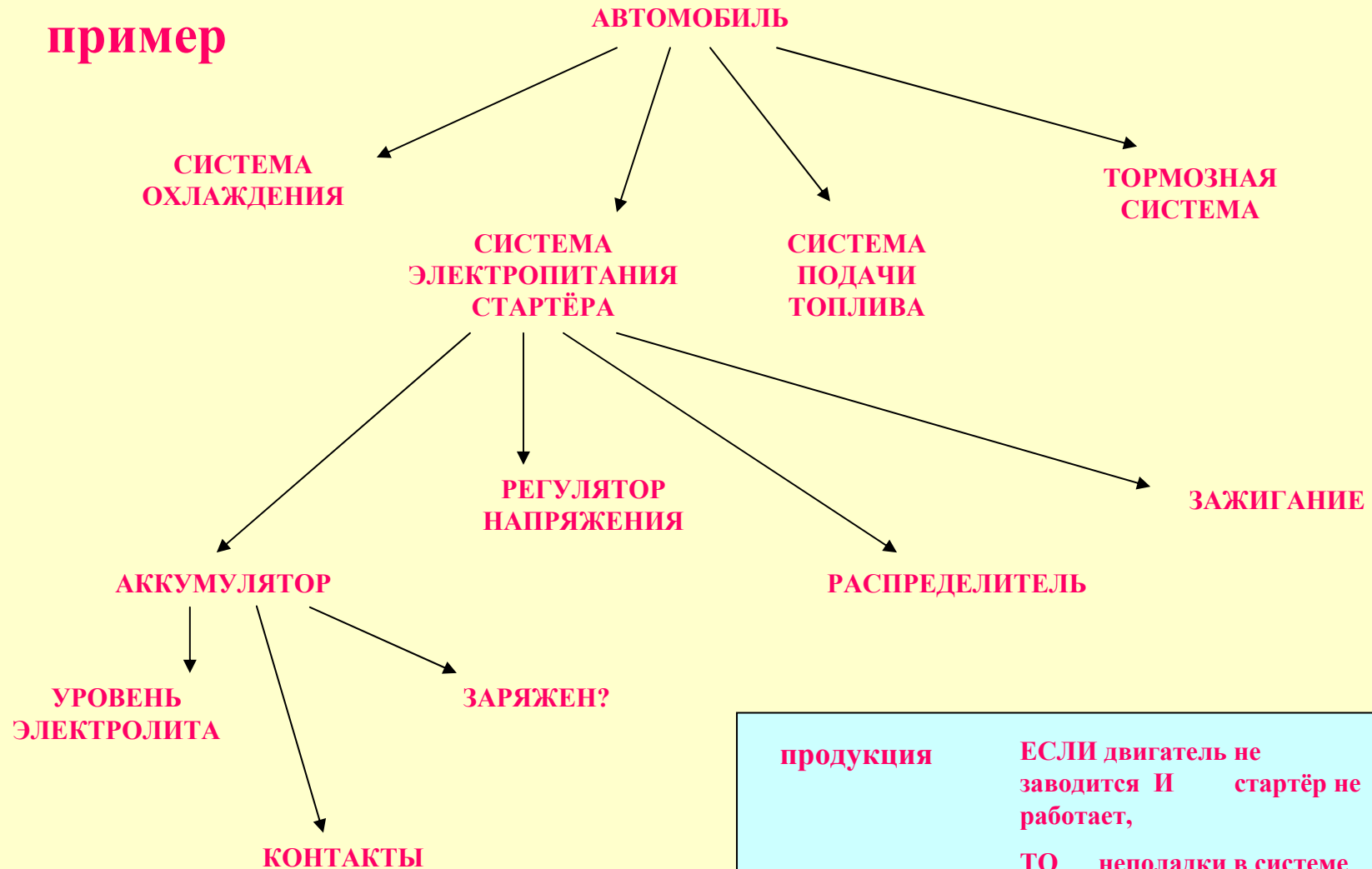
# Структура экспертной системы

(2)



# Пример ЭС

**пример**



**продукция**

**ЕСЛИ** двигатель не заводится **И** стартёр не работает,

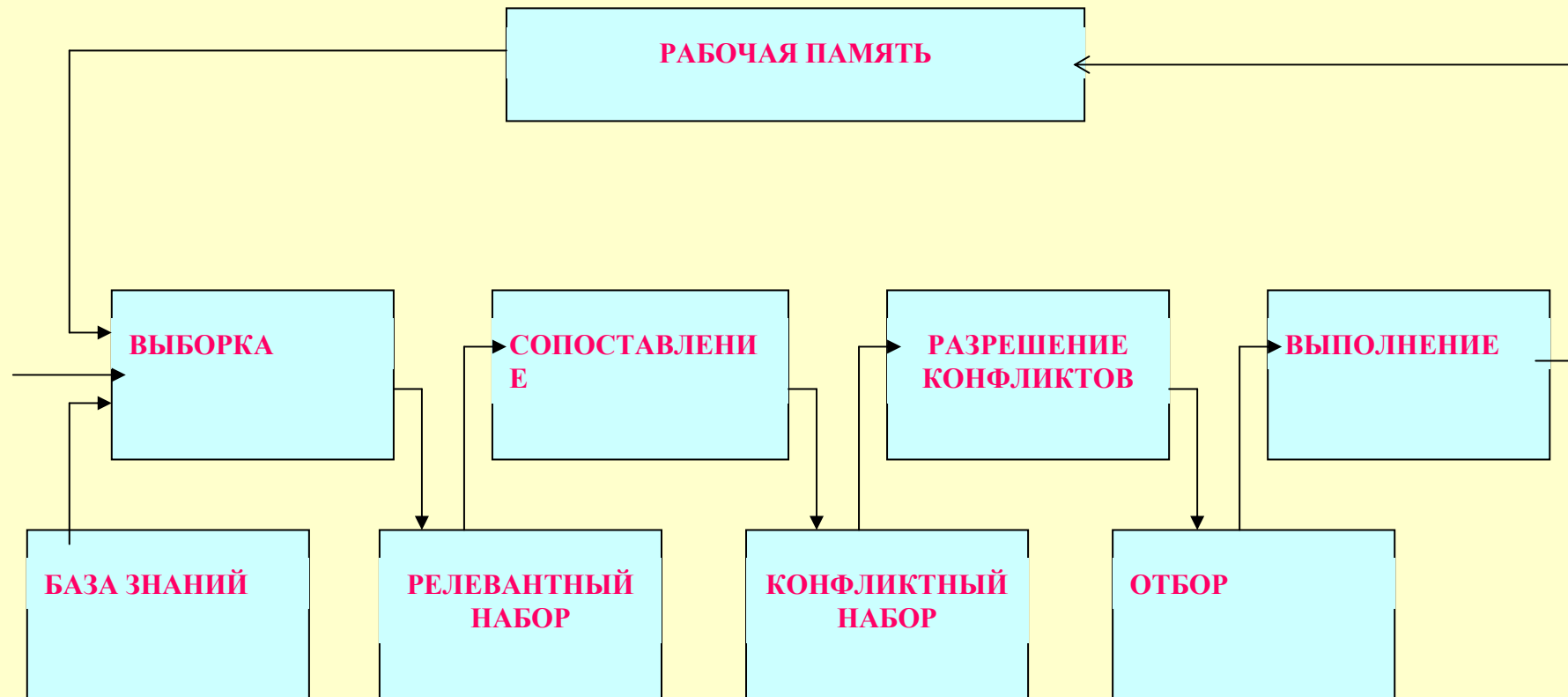
**ТО** неполадки в системе электропитания стартёра

# Пример работы ЭС «Диагностика двигателя» с подсистемой объяснений

- **Система:** Имеется ли бензин в топливном баке?  
**Пользователь:** Да.  
**Система:** Истинно ли то, что двигатель работает ?  
**Пользователь :** Нет  
**Система:** Истинно ли, что фары горят ?  
**Пользователь:** Почему?  
**Система:** Я пытаюсь доказать, что проблема с аккумулятором. Было установлено, что двигатель не работает. Поэтому, если не имеет место, что фары горят, тогда имеется проблема с аккумулятором.  
Истинно ли, что фары горят?  
**Пользователь:** Нет  
**Система:** я заключаю, что имеется проблема с аккумулятором.  
**Пользователь:** Как?  
**Система:** Это следует из правила 2:  
IF NOT engine\_turns\_over  
AND NOT engine\_turns\_overwas  
- ответ, данный пользователем.  
NOT lights\_come\_onwas - ответ, данный пользователем.

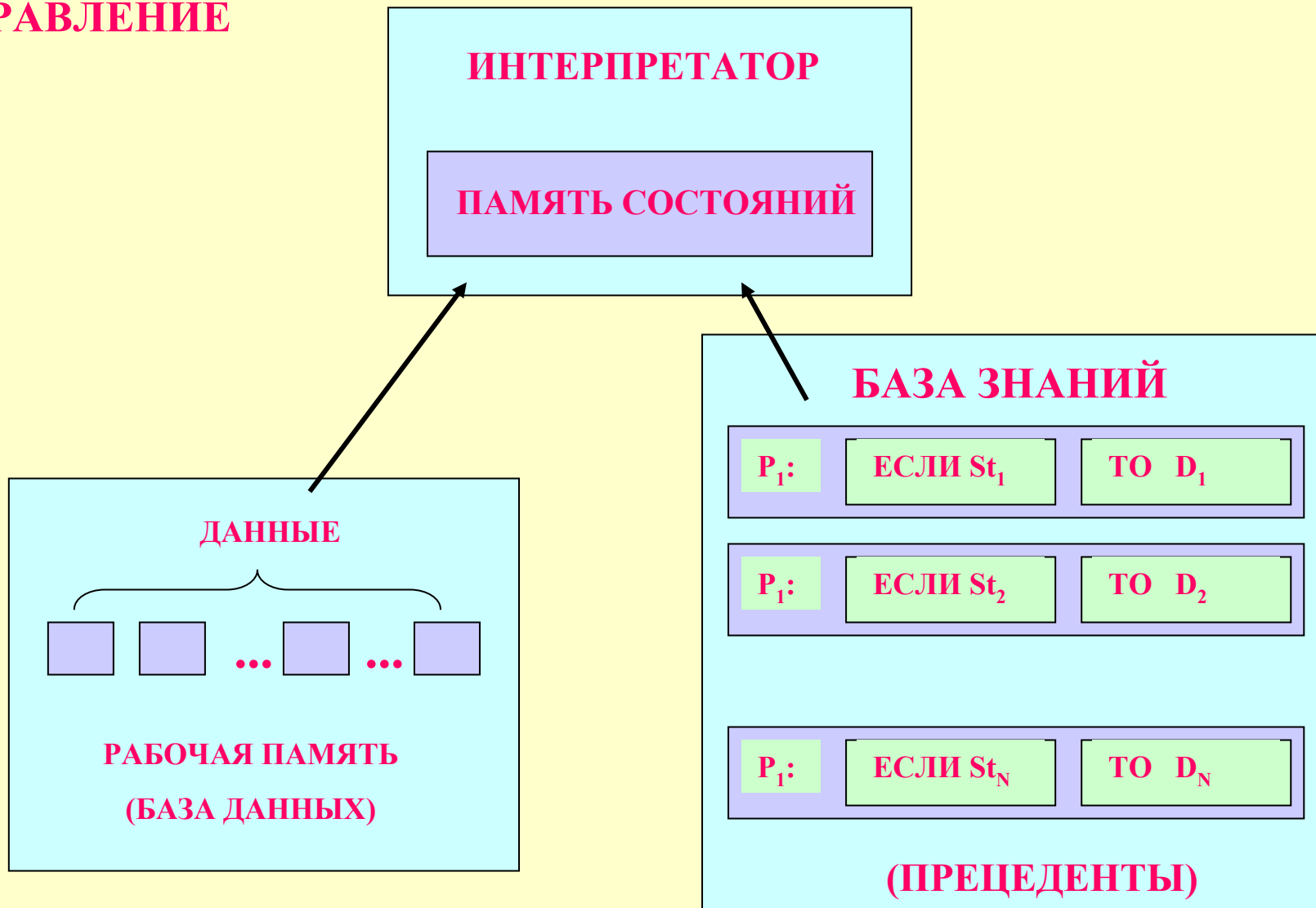
# Цикл работы экспертной системы

## ЦИКЛ РАБОТЫ



# Управление в ЭС

**УПРАВЛЕНИЕ**



# Прототипы ЭС (стадии существования ЭС)

- Демонстрационный
  - Для демонстрации возможностей выбранных архитектурных решений (может отсутствовать при использовании действующего прототипа)
- Исследовательский
  - Для совершенствования архитектуры ЭС и базы знаний
- Действующий
  - Лишенный незавершенности, свойственной исследовательскому прототипу (иногда вместо него подразумевают исследовательский)
- Промышленный
  - Для реального применения в рамках одной организации/корпорации (для которой разрабатывалась ЭС)
- Коммерческий
  - Для широкого применения во многих организациях/корпорациях



# Исследовательский прототип

- *Исследовательским прототипом* называют систему, которая решает представительный класс задач приложения, но может быть неустойчива в работе и не полностью проверена. При наличии развитых инструментальных средств (ИС) для разработки исследовательского прототипа требуется примерно 2 - 4 месяца. Исследовательский прототип обычно имеет в базе знаний не больше 50 общих исполняемых утверждений; при использовании только частных утверждений их количество возрастает в 3 - 10 раз.

# Действующий прототип

- *Действующий прототип* надежно решает все задачи, но для решения сложных задач может требовать чрезмерно много времени и (или) памяти. Доведение системы от начала разработки до стадии действующего прототипа требует примерно 6 - 9 месяцев, при этом количество исполняемых утверждений в базе знаний увеличивается до 100.

# Промышленный прототип

- ЭС, достигшая стадии *промышленной системы*, обеспечивает высокое качество решений всех задач при минимуме времени и памяти. Обычно процесс преобразования действующего прототипа в промышленную систему состоит в расширении базы знаний (до 150 исполняемых утверждений) и ее тщательной отладке. Доведение ЭС от начала разработки до стадии промышленной системы на развитом ИС требует примерно 12 - 18 месяцев.

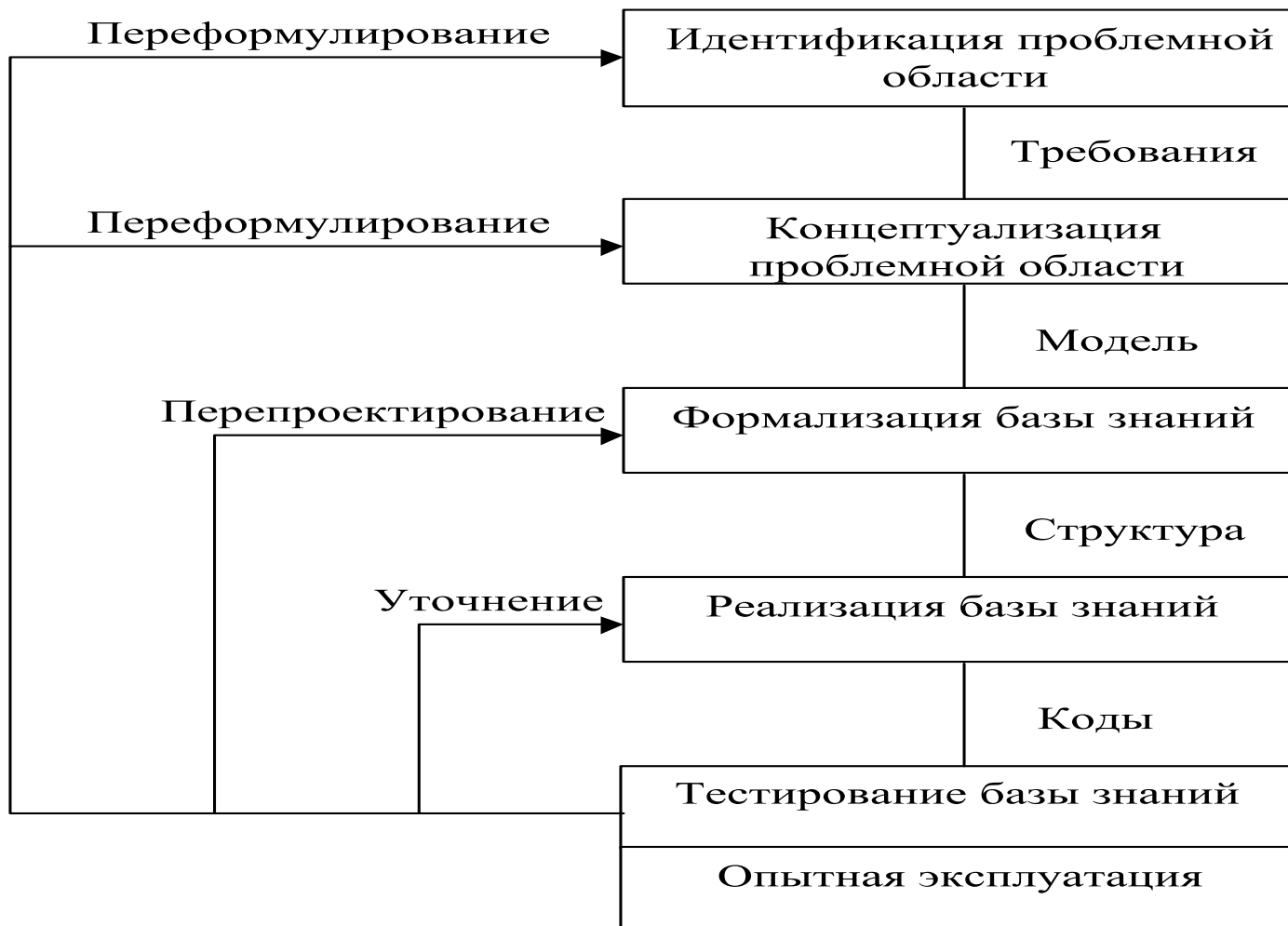
# Коммерческий прототип

- Обобщение задач, решаемых ЭС на стадии промышленной системы, позволяет перейти к стадии *коммерческой системы*, т.е. к системе, пригодной не только для собственного использования, но и для продажи различным потребителям. Доведение системы до коммерческой стадии требует примерно 1,5 - 2 года. Приведенные выше сроки справедливы для ЭС средней сложности.
- Note: *приведенные выше количественные оценки в разных источниках могут быть разные.*

# Коллектив разработчиков ЭС

- Эксперт
  - Предоставление знаний для построения ЭС
- Инженер по знаниям
  - Извлечение и формализация знаний
  - Разработка или выбор архитектуры ЭС
  - Разработка процедур тестирования
- Программист
  - Разработка и кодирование компонентов ЭС или дополнений к выбранному инструментальному средству
- Пользователь
  - Использование ЭС (решение задач предметной области)
- Тестировщик
  - Тестирование прототипов ЭС
- Руководитель проекта
  - Координация работ исполнителей

# Этапы создания ЭС



# Идентификация

Приемы	Описание
1. Наблюдение	Инженер наблюдает, не вмешиваясь, за тем, как эксперт решает реальную задачу
2. Обсуждение задачи	Инженер на представительном множестве задач неформально обсуждает с экспертом данные, знания и процедуры решения
3. Описание задачи	Эксперт описывает решение задач для типичных запросов
4. Анализ решения	Эксперт комментирует получаемые результаты решения задачи, детализируя ход рассуждений
5. Проверка системы	Эксперт предлагает инженеру перечень задач для решения (от простых до сложных), которые решаются разработанной системой
6. Исследование системы	Эксперт исследует и критикует структуру базы знаний и работу механизма вывода
7. Оценка системы	Инженер предлагает новым экспертам оценить решения разработанной системы

# 1. Идентификация

1.1. Определение участников и их ролей в процессе создания и эксплуатации экспертной системы

1.2. Идентификация проблемы

1.3. Определение необходимых ресурсов - временных, людских, материальных

1.4. Определение целей

В качестве целей, преследуемых при создании экспертных систем, могут быть: повышение скорости принятия решения, повышение качества решений, тиражирование опыта экспертов и т.п.



# 1.2. Идентификация проблемы

- Какой класс задач должна решать ЭС
- Как эти задачи могут быть охарактеризованы или определены
- Какие можно выделить подзадачи
- Какие исходные данные должны использоваться для решения
- Какие понятия и взаимосвязи между ними используются при решении задачи экспертами
- Какой вид имеет решение и какие концепции используются в нем
- Какие аспекты опыта эксперта существенны для решения задачи
- Какова природа и объем знаний, необходимых для решения задачи
- Какие препятствия встречаются при решении задач
- Как эти помехи могут влиять на решение задачи

## 2. Концептуализация

- Какие типы данных нужно использовать
- Что из данных задано, а что должно быть выведено
- Имеют ли подзадачи наименования
- Имеют ли стратегии наименования
- Имеются ли ясные частичные гипотезы, которые широко используются

# Этапы разработки ЭС

Этап разработки	Характер прототипа	Количество правил	Срок разработки	Стоимость
Идентификация	Демонстрацион-ный	50 - 100	1 - 2 мес.	
Концептуализация	Исследователь-ский	200 - 500	3 - 6 мес.	25 - 50т.\$
Формализация				
Реализация	Действующий	500 - 1000	6 - 12 мес.	
Тестирование	Промышленный	1000 - 1500	1 - 1,5 года	300т.\$
Опытная эксплуатация	Коммерческий	1500 - 3000	1,5 - 3 года	2 - 5 млн.\$

# Стратегии разработки

	Широкий набор задач	Концентрированный набор задач	Комплексный набор задач
Назначение	Автоматизация	Стандартизация, повышение качества	Реорганизация бизнес-процессов
Требования к разработчикам	Эксперты-пользователи	Профессиональные команды	Междисциплинарные команды
Стоимость	Низкая на проект	Высокая на проект	Высокая на проект
Риск	Диверсифицированный	Концентрированный	Концентрированный
Примеры	DuPont du Nemours Оболочка Insight Plus Сотни экспертных систем. Сотни правил в каждой ЭС	DEC, ЭС конфигурирования компьютеров XCON, продажи XSEL 17000 правил, эффект 27 млн. долл.	Xerox Среда разработки информационной системы ART-Enterprise (Inference) Интеллект. моделирование ReThink (Gensym)

# Средства разработки ЭС

1. Традиционные (в том числе объектно-ориентированные) языки программирования типа С, С++ (как правило, эти ИС используются не для создания ЭС, а для создания ИС).
2. Символьные языки программирования (например, Lisp, Prolog и их разновидности). Эти ИС в последнее время, как правило, не используются в реальных приложениях в связи с тем, что они плохо приспособлены к объединению с программами, написанными на языках традиционного программирования.
3. Инструментарий, содержащий многие, но не все компоненты ЭС. Эти средства предназначены для разработчика, от которого требуются знание программирования и умение интегрировать компоненты в программный комплекс. Примерами являются такие средства, как OPS 5, ИЛИС и др.
4. Оболочки ЭС общего назначения, содержащие все программные компоненты, но не имеющие знаний о конкретных предметных средах. Средства этого и последующего типов не требуют от разработчика приложения знания программирования. Примерами являются ЭКО, Leonardo, Nexpert Object, Каппа и др.

Подчеркнем, что в последнее время термин "оболочка" (shell) используется реже, его заменяют на более широкий термин "среда разработки" (development environment). Если хотят подчеркнуть, что средство используется не только на стадии разработки приложения, но и на стадиях использования и сопровождения, то употребляют термин "полная среда" (complete environment). Примерами таких средств для создания статических ЭС являются: Nexpert Object, ProKappa, ART\*Enterprise, Level 5 Object и др.

# Средства разработки ЭС (2)

5. Проблемно/предметно-ориентированные оболочки (среды):

- проблемно-ориентированные средства (problem-specific), ориентированные на некоторый класс решаемых задач и имеющие в своем составе соответствующие этому классу альтернативные функциональные модули (примерами таких классов задач являются задачи поиска, управления, планирования, прогнозирования и т.п.);
- предметно-ориентированные средства (domain-specific), включающие знания о некоторых типах предметных областей, что сокращает время разработки БЗ.

При использовании инструментария четвертого типа могут возникнуть следующие трудности:

- 1) управляющие стратегии, вложенные в механизм вывода инструментария, могут не соответствовать методам решения, которые использует эксперт, взаимодействующий с данной системой, что может привести к неэффективным, а возможно, и неправильным решениям;
- 2) язык представления знаний, принятый в инструментарии, может не подходить для данного приложения.

Значительная компенсация этих трудностей достигается применением проблемно/предметно-ориентированных средств (ИС пятого типа).

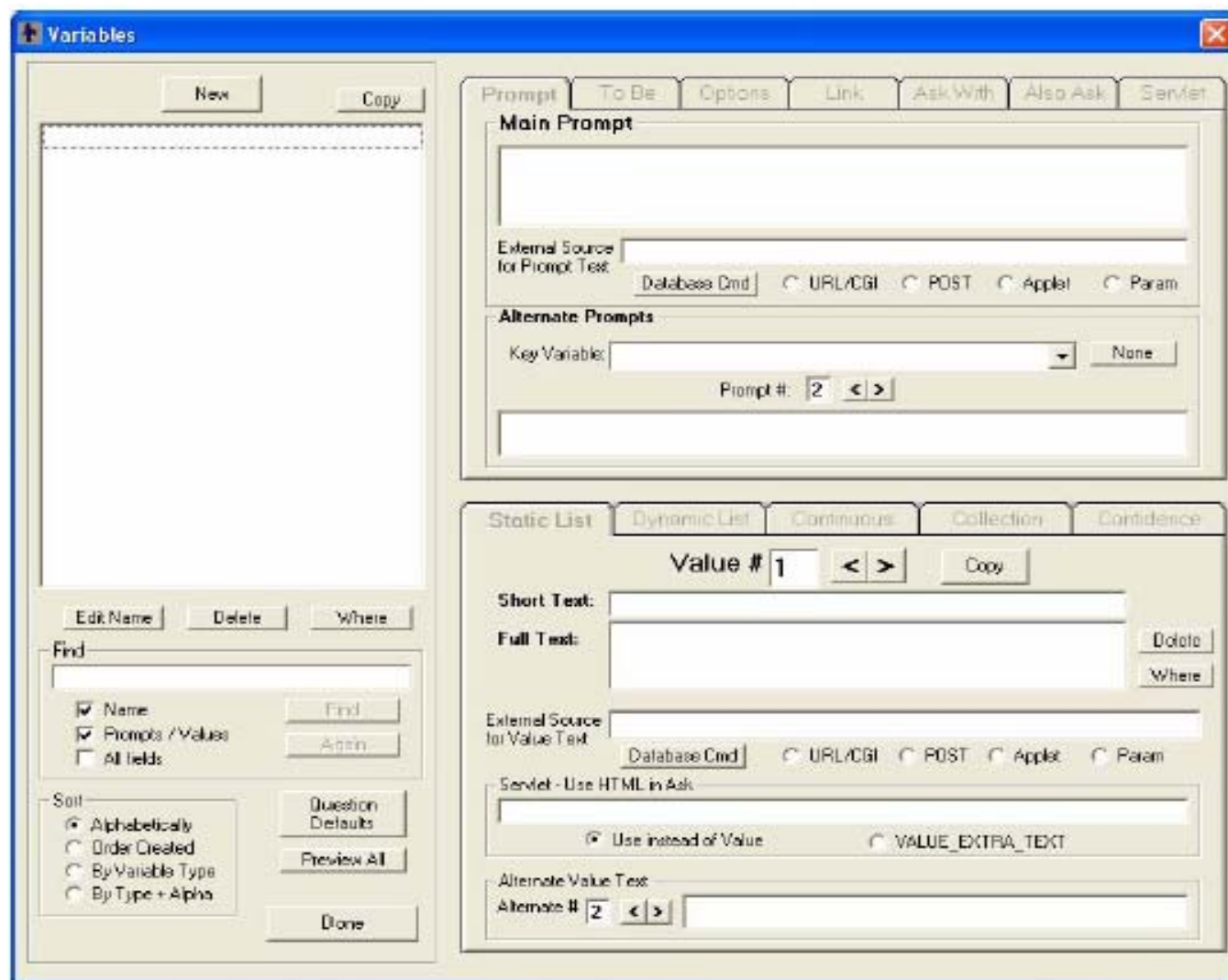
Описание метода представления знаний	Программные инструментальные средства						
	ЭКО ArgusSoft	GURU MDBS	Nexpert Object Neuron Data	LEVEL-5 Level Co	ART Enterprise Inference	G2 Gensym	ESW in 2.0
Описание объектов:							
Семант. сеть						*	
Объекты (Фреймы)			*	*	*	*	*
Предикаты						*	
Правила	*	*	*	*	*	*	*
Действия:							
Правила	*	*	*	*	*	*	*
Сообщения			*	*	*		*
Предикаты						*	
Процедуры	*	*	*	*	*	*	
Неопределенность:							
Наслед. свойств			*	*	*	*	*
Коэф. уверен.		*		*	*	*	*
Неч. множества		*		*	*		*
Усл. вероятн.	*						
Вывод:							
Прямая цепочка		*	*	*	*	*	
Обратн. цепочка	*	*	*	*	*	*	*
Объект. ориент.			*	*	*	*	
Гипотет. вывод					*	*	
Обраб. времени				*		*	
Доска объявл.						*	

# Наиболее популярные ИС для ЭС (США)

- *Малые ИС:*
  - VP Expert (paperback Software),
  - 1st Class (1st Class Expert System).
  - Personal Consultant Easy,
  - Procedure Consultant,
  - Crystal,
  - Exsys Corvid.
- *Средние ИС:*
  - Nexpert Object (Neuron Data),
  - ProKappa (Intellicorp),
  - Art-IM,
  - ART Enterprise (Inference),
  - level 5 Object (IBI).
- *Большие ИС:*
  - Aion DS 5.1 (Trinzic).
  - KBMS (Trinzic),
  - ART (Inference).
- *Символьные ИС:*
  - KEE (Intellicorp),
  - ART (Inference).
  - Gold Works (Golden Hill),
  - Mercury (AIT).



# Оболочка EXSYS CORVID



Окно для определения переменных

А.В.Гаврилов  
НГТУ, кафедра ВТ

# Оболочка EXSYS CORVID (2)

**New Variable:**

Name:

Type:

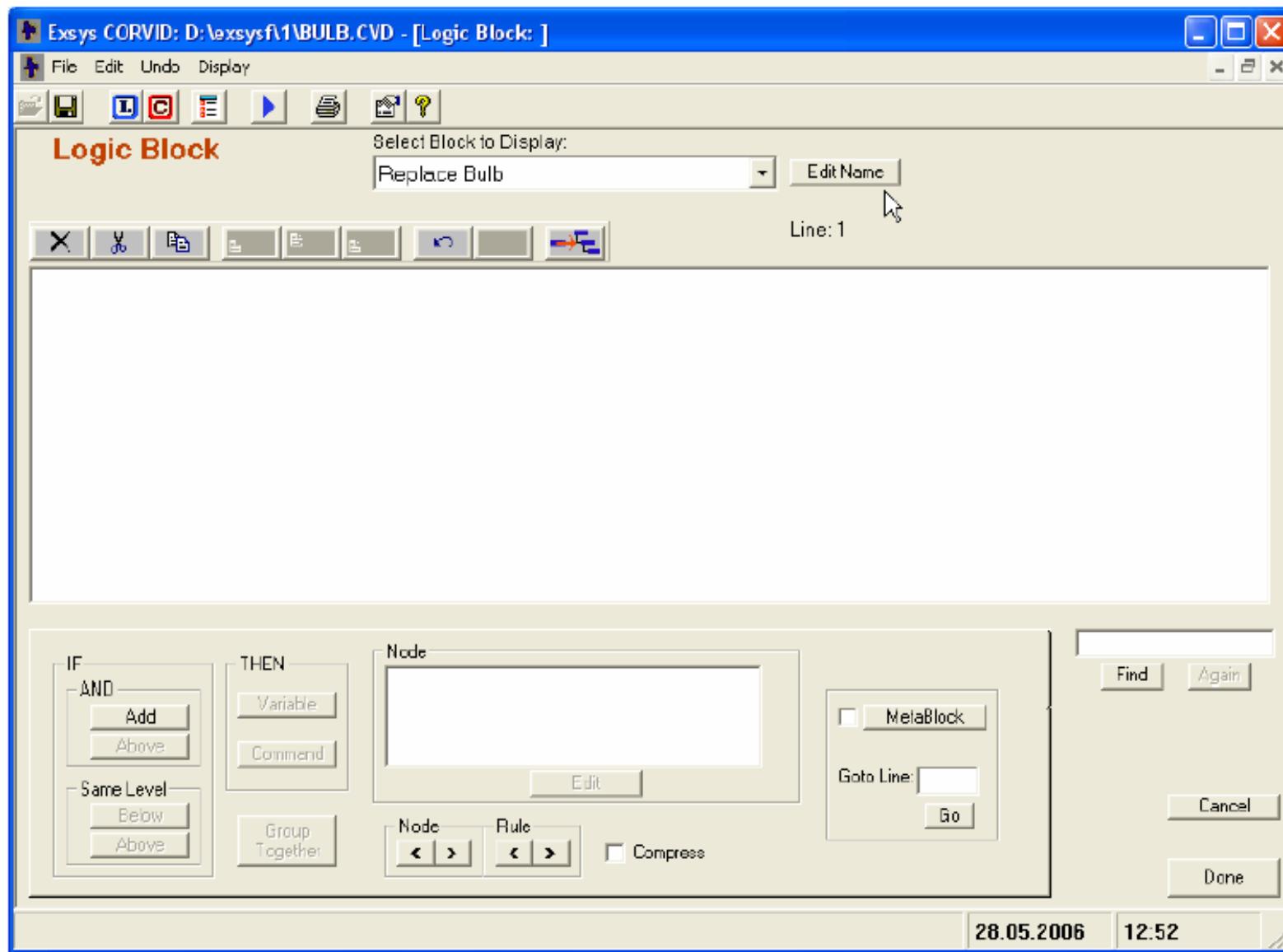
- Static List - Predefined list of values
- Dynamic List - List of values set at Runtime
- Numeric value
- String value
- Date value
- Collection - Value is a group of items added dynamically
- Confidence - Value will be a confidence factor

Cancel

OK

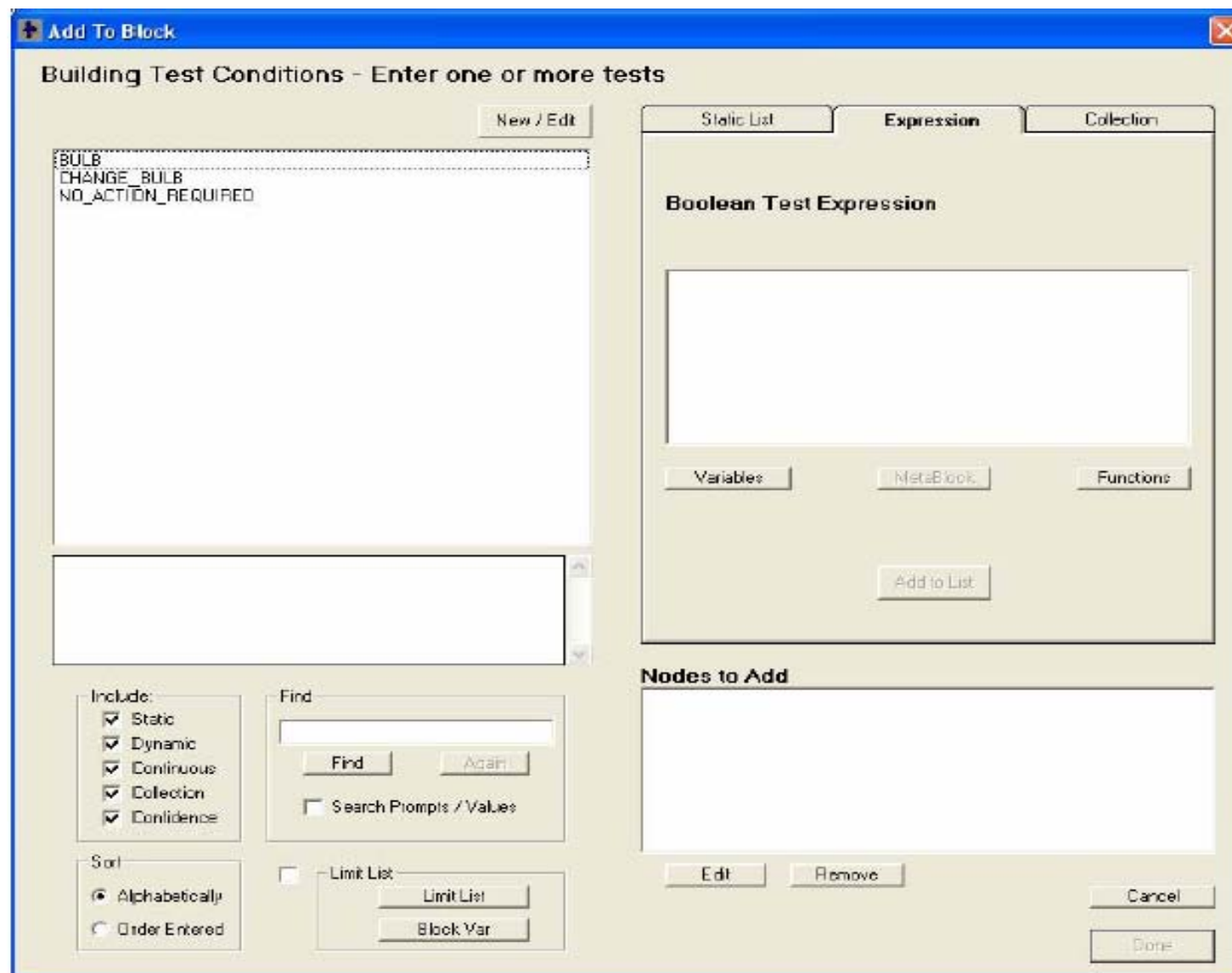
Окно для выбора типа переменной

# Оболочка EXSYS CORVID (3)



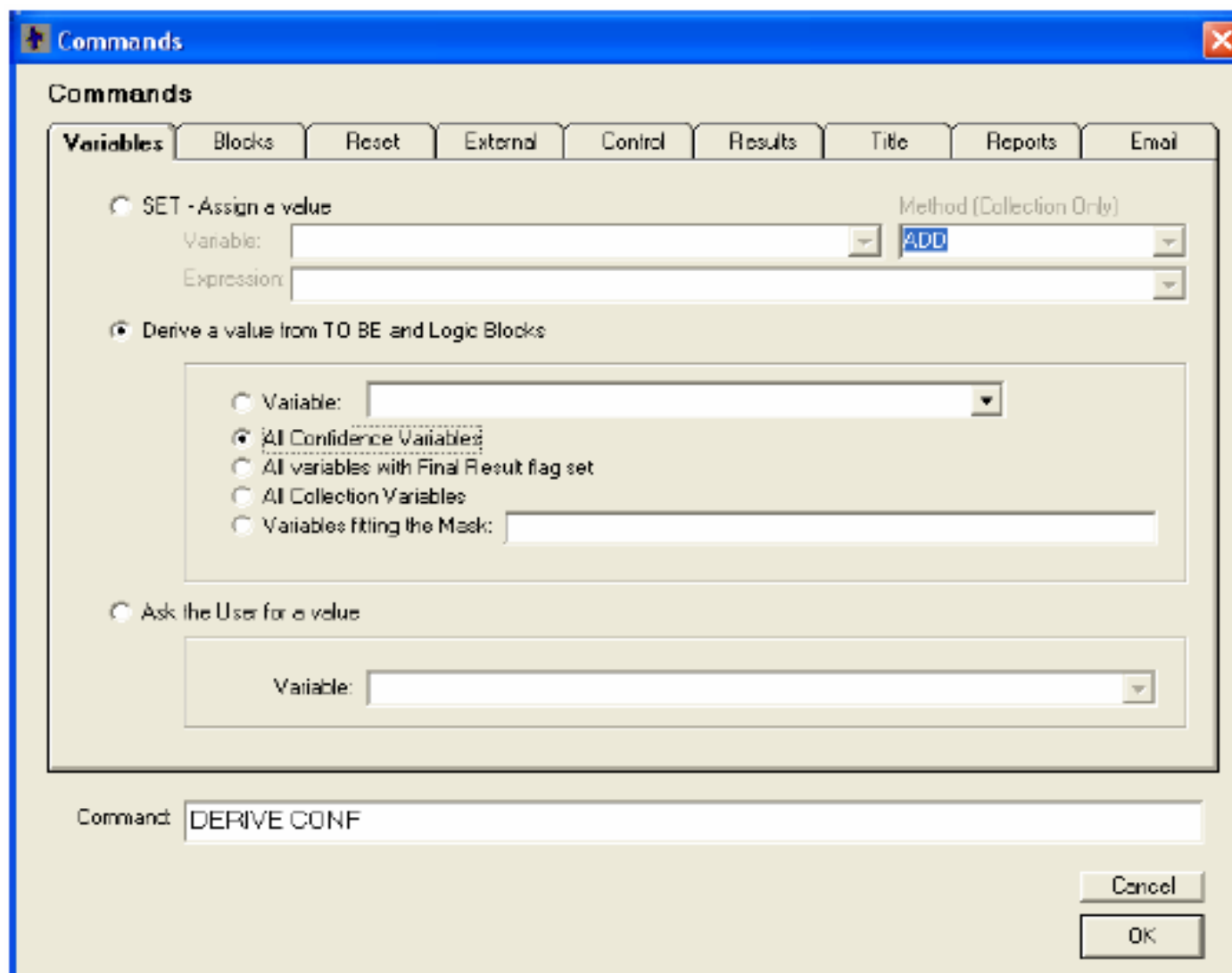
А.В.Гаврилов  
НГТУ, кафедра ВТ

# Оболочка EXSYS CORVID (4)



А.В.Гаврилов  
НГТУ, кафедра ВТ

# Оболочка EXSYS CORVID (5)



# Малые ИС для ЭС, разработанные в России

- ЭКО,
- МОДИС,
- SIMER+MIR,
- ЭКСПЕРТ,
- ЛЭДИ,
- МЭС,
- ЭСПЛАН,
- ФИАКР,
- ПИЭС,
- ЗНАТОК,
- ЭКРАН,
- ШЭДЛ
- ESWin

# Инструментальное ПО ESWin

## **В состав инструментального ПО входят:**

- экспертная оболочка для запуска экспертных систем ESWin с целью их отладки разработчиком экспертных систем,
- интерпретатор баз знаний ESWinUs для запуска экспертных систем конечным пользователем,
- редакторы баз знаний EdKB и KlbEdit,
- программа для просмотра и диагностики целостности баз знаний KBView,
- программа для редактирования и оптимизации баз знаний KBOptim

# Инструментальное ПО ESWin (2)

**В качестве методов представления знаний использованы:**

- правила-продукции с представлением нечеткости в виде коэффициентов достоверности с обратным логическим выводом,
- фреймы для описания структуры предметной области и диалога с пользователем,
- лингвистические переменные для описания нечетких понятий, входящих во фреймы.



# Инструментальное ПО ESWin (3)

**ПО поддерживает решение задач** методом обратного нечеткого логического вывода. При этом факты берутся из диалога с пользователем и из баз данных через SQL-запросы.

В диалоге можно использовать для пояснения графику в формате BMP, GIF, AVI, HTML, при проверке условий и формировании значений в заключениях правил можно использовать простые арифметические выражения.

В процессе решения задачи используются различные виды контекста, в том числе, наследование, что облегчает написание баз знаний, и делает их достаточно гибкими.

Можно использовать при описании предметной области отношения между понятиями типа "часть-целое" и "элемент-класс".

# Оболочка ESWin

## Пример БЗ

```
TITLE = для выбора метода представления знаний
FRAME = Цель
    Метод представления знаний: ()
ENDF
FRAME = Тип
    Решаемые задачи: (диагностика; проектирование)
ENDF
FRAME = Область
    Применение [Какова область применения?]: (медицина; вычислительная техника)
ENDF
FRAME = Количество
    Число правил в базе знаний (численный): ()
    Число объектов в базе знаний (численный): ()
ENDF
FRAME = Действие
    Сообщение: ()
ENDF
RULE 1
    > (Количество.Число правил в базе знаний; 50)
    < (Количество.Число правил в базе знаний; 100)
    < (Количество.Число объектов в базе знаний; 30)
DO
    = (Тип.Решаемые задачи; диагностика) 100
ENDR
```

```
RULE 2
    > (Количество.Число правил в базе знаний; 100)
    > (Количество.Число объектов в базе знаний; 30)
DO
    = (Тип.Решаемые задачи; проектирование) 100
ENDR
RULE 3
    = (Область.Применение; медицина)
    = (Тип.Решаемые задачи; диагностика)
DO
    = (Метод представления знаний; Правила-продукции с представлением нечетких знаний) 90
ENDR
RULE 4
    = (Область.Применение; вычислительная техника)
    = (Тип.Решаемые задачи; проектирование)
DO
    = (Метод представления знаний; Фреймы) 100
    = (Метод представления знаний; Правила-продукции с представлением нечетких знаний) 70
    = (Метод представления знаний; Семантические сети) 70
    MS(Действие.Сообщение; Доказано правило 4)
ENDR
```

# Оболочка ESWin (2)

Отношения в условиях могут быть:

EQ| = равно;

GT| > больше;

LT| < меньше;

NE|  $\neq$  не равно;

IN два фрейма связаны отношением "часть-целое" (имеется связь через слот OWNER).

В случае символьных слотов может использоваться только условия EQ и NE. В случае лингвистической переменной - все арифметические отношения (EQ, GT, LT, NE). Если значением слота является слово "any", это означает, что данное условие всегда истинно и используется только для запроса значения слота.

Отношения в заключениях могут быть:

EQ| = равно (создание факта - слота во фрейме-экземпляре);

IN включение во фрейм-владелец (создание связи - слота OWNER во фрейме-экземпляре);

DL удаление слота во фрейме-экземпляре;

EX запуск внешней программы;

FR вывод фрейма-экземпляра;

GO запуск правила;

MS вывод на экран сообщения;

GR вывод на экран графического файла (форматов \*.gif, \*.avi или \*.htm).

# Оболочка ESWin (3)

Пример конструкции SOURCE описания доступа к базе данных с помощью конструкции SOURCE:

```
Frame=Демо для ESWin
```

```
Parent:
```

```
Table: ("Запрос1")
```

```
СУБД: (Access 97)
```

```
База данных: (Борей)
```

```
EndF
```

```
Frame=Товар
```

```
Parent:
```

```
Тип:
```

```
Марка:
```

```
Цена(численный): ()
```

```
На складе(численный): ()
```

```
EndF
```

```
Source=Товар
```

```
Parent: Демо для ESWin
```

```
Тип: (Категория)
```

```
Марка: (Марка)
```

```
Цена(численный): (Цена)
```

```
На складе(численный): (НаСкладе)
```

```
EndS
```

Этот пример описывает доступ к демонстрационной базе данных БОРЕЙ,

поставляемой с СУБД Access А.В.Гаврилов

НГТУ, кафедра ВТ

# Оболочка ESWin (4)

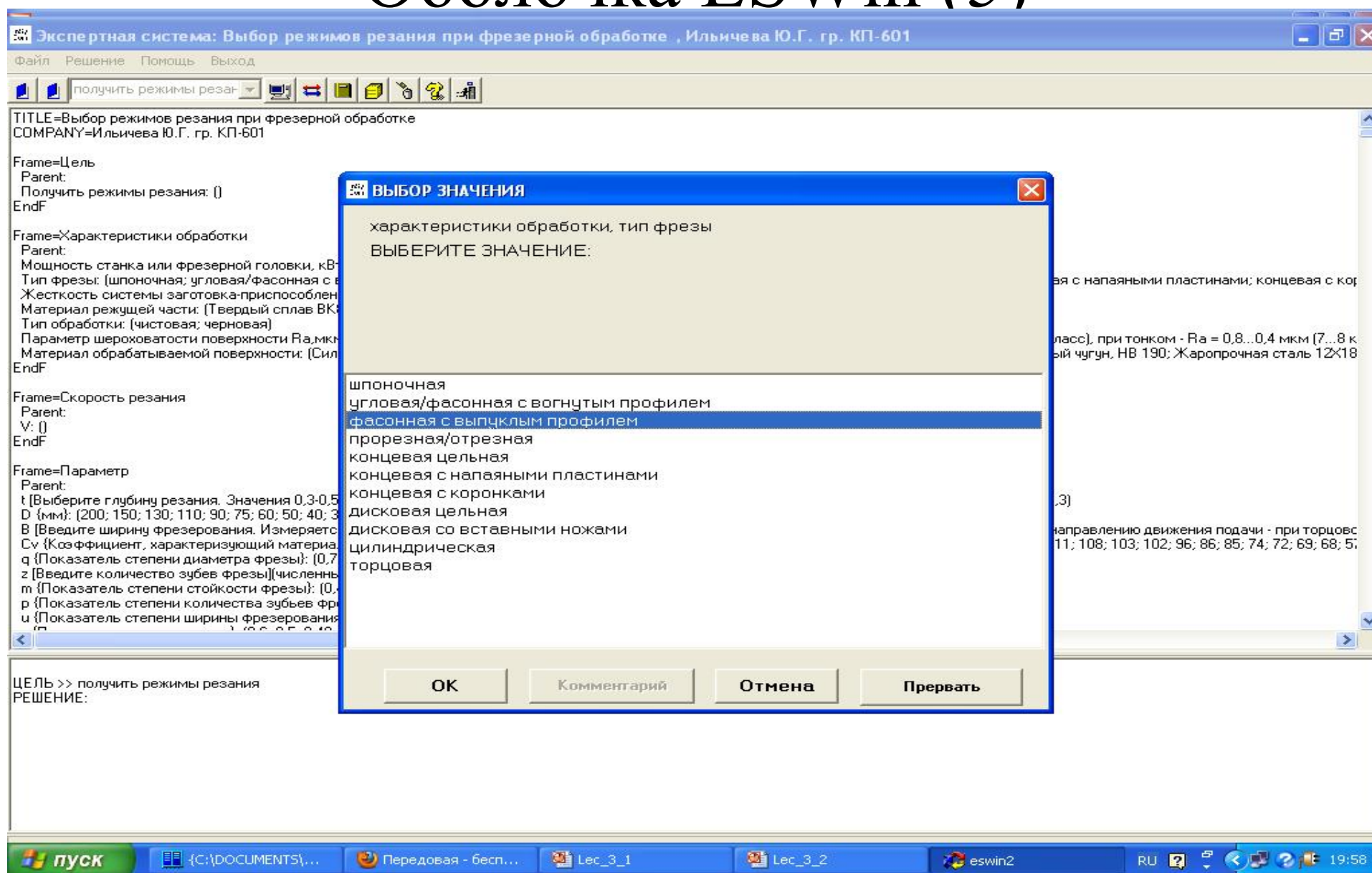
При проверке условия при интерпретации правила

Определение значения слота производится в следующем порядке:

- 1) ищется факт со значением этого слота в базе фактов;
- 2) ищется значение слота в одноименном фрейме-классе (если это значение в нем единственное);
- 3) ищется правило, задающее при срабатывании значение данного слота, и запускается его интерпретация;
- 4) ищется одноименная структура SOURCE для формирования SQL-запроса к внешней базе данных;
- 5) ищется слот во фрейме-классе с описанием всей необходимой информации для запроса значения слота у пользователя.

К каждому из этих шагов оболочка переходит в случае неудачи предыдущего шага.

# Оболочка ESWin (5)



А.В.Гаврилов  
НГТУ, кафедра ВТ

# Редактор EdKb (2)

Окно для редактирования фрейма

**Фрейм: Область** [X]

Удалить фрейм < >

Тип фрейма: Класс

Слоты:

- Parent
- Применение**

Имя нового слота: Применение

Значение слота:

Значения слота (меню):

- медицина
- вычислительная техника

символьный

Вопрос к слоту: Какова область применения?

Удалить слот    Удалить значение

Добавить слот    Добавить значение

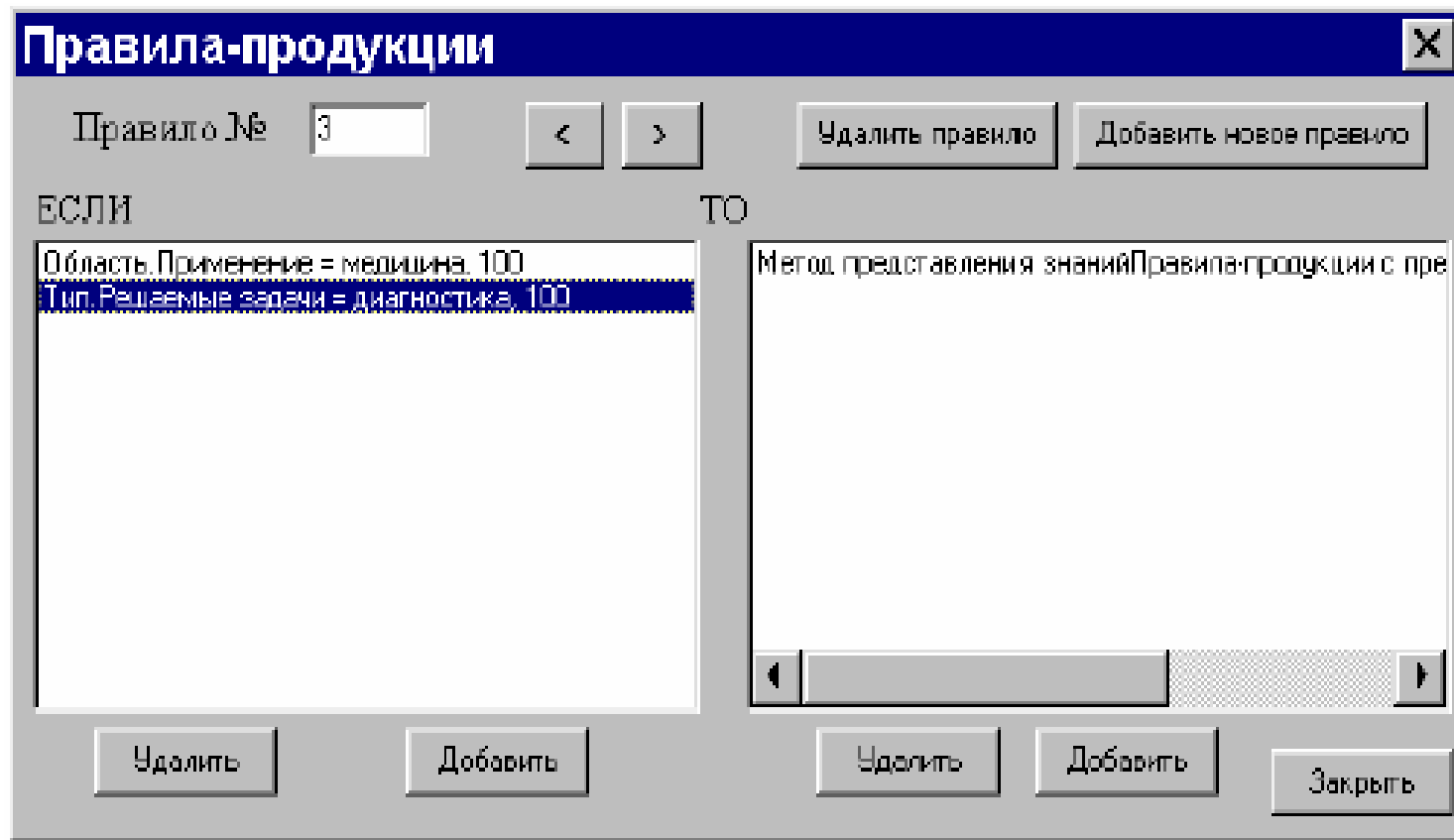
Выбор фрейма в качестве имени(значения):

Замена    ОК    Закрыть

А.В.Гаврилов  
НГТУ, кафедра ВТ

# Редактор EdKb (3)

Окно для редактирования правил





# Редактор EdKb (4)

Окно для редактирования условия/заключения в правиле

Формирование условия (заключения)

Фрейм.слот      Отношение      Значение

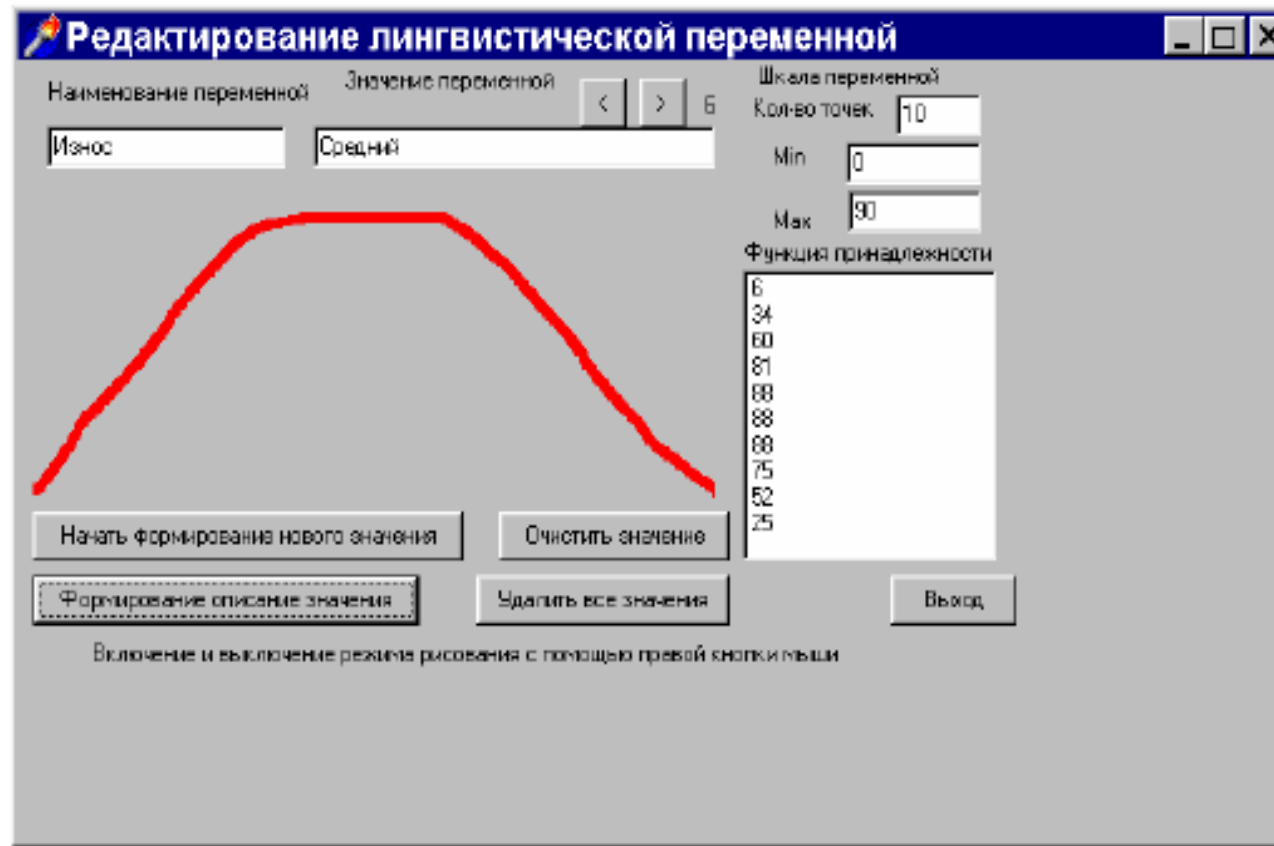
Тип. Решаемые задачи      =      проектирование

Цель	Решаемые задачи	диагностика
Тип		проектирование
Область		
Количество		
Действие		

Коэфф. дост. 100      Фрейм      Отмена      ОК

# Редактор EdKb (5)

Окно для редактирования значения лингвистической переменной



А.В.Гаврилов  
НГТУ, кафедра ВТ

# Редактор EdKb (6)

Окно для редактирования значения лингвистической переменной

Редактирование лингвистической переменной

Наименование переменной: Износ

Значение переменной: Незначительный

Шкала переменной: Кол-во точек: 10, Min: 0, Max: 90

Функция принадлежности: 97, 91, 34, 4, 2, 2, 2, 2, 2, 2

Начать формирование нового значения, Очистить значение, Формирование описание значения, Удалить все значения, Выход

Включение и выключение режима рисования с помощью правой кнопки мыши

# Средние ИС российского производства

- НЭКС,
- ЭКСПЕРТ-МИКРО,
- ЭКСНА,
- КОНС-ПРОЛОГ,
- ПРОДУС

# Большие ИС российского производства

- НЭКС,
- ОПС-86,
- МЕДИФОР,
- КОМФОРТ,
- КОНСУЛЬТАНТ-2,
- РЕЛЯП

# Подходы к разработке ЭС

1. подход, базирующийся на поверхностных знаниях;
2. структурный подход;
3. подход, базирующийся на глубинных знаниях;
4. смешанный подход, базирующийся на использовании поверхностных и глубинных знаний.

- ***Подход, базирующийся на поверхностных знаниях***, применяется к сложным задачам, которые не могут быть точно описаны. Этот подход заключается в получении от эксперта фрагментов знаний (часто звристических), которые релевантны решаемой задаче, При этом не предпринимается никаких попыток систематического или глубинного изучения области, что предопределяет использование поиска в пространстве состояний в качестве универсального механизма вывода. Обычно в ЭС, использующих данный подход, в качестве способа представления выбираются правила.

- *Структурный подход* к построению ЭС обусловлен тем, что для ряда приложений применение только техники поверхностных знаний не обеспечивает решения задачи. Действительно, использование поиска в качестве механизма вывода в неструктурированной базе знаний может приводить к ненадежным и (или) некачественным решениям. Структурный подход к построению ЭС подобен структурному программированию. Однако применительно к ЭС речь не идет о том, что структурирование должно довести задачу до алгоритма (как в традиционном программировании), а предполагается, что часть задачи решается с помощью поиска. Структурный подход в различных приложениях целесообразно сочетать с поверхностным или глубинным.



- В *глубинном подходе* компетентность ЭС базируется на модели той проблемной среды, в которой эта ЭС работает. Модель может быть определена различными способами (декларативно, процедурно). Необходимость в ряде приложений использовать модели вызвана стремлением исправить несовершенство поверхностного подхода, возникающего при отсутствии правил, удовлетворяющих текущей ситуации в рабочей памяти. Глубинные ЭС кроме возможностей поверхностных ЭС обладают способностью при возникновении неизвестной ситуации определить с помощью некоторых общих принципов, справедливых для области экспертизы, какие действия следует выполнить.

- *Смешанный подход* в общем случае может сочетать поверхностный, структурный и глубинный подходы. Например, поверхностный подход может быть использован для поиска адекватных знаний, которые затем используются некоторой глубинной моделью.

# Задача приобретения знаний. Характеристики ИС в контексте приобретения знаний

1. Уровень языка, в котором осуществляется приобретение знаний:
  - формализованный язык;
  - ограниченный естественный язык (ЕЯ);
  - язык пиктограмм и изображений;
  - ЕЯ и язык изображений

## 2. Тип приобретаемых знаний:

- данные в виде таблиц, содержащих значения входных и выходного атрибутов (по которым индуктивными методами строится дерево вывода);
- специализированные правила;
- общие и специализированные правила.

## 3. Тип приобретаемых данных:

- атрибуты со значениями;
- объекты;
- классы структурированных объектов и их экземпляры, получающие значения атрибутов по наследованию

# Методы извлечения знаний



# Классификация вопросов при интервьюировании



# Мозговой штурм

- Метод «*мозгового штурма*» - один из наиболее известных и широко применяемых методов генерирования новых идей путем творческого сотрудничества группы специалистов. Являясь в некотором смысле единым мозгом, группа пытается штурмом преодолеть трудности, мешающие разрешить рассматриваемую проблему. В процессе такого штурма участники выдвигают и развивают собственные идеи, стимулируя появление новых и комбинируя их. Для обеспечения максимального эффекта «мозговой штурм» должен подчиняться определенным правилам и основываться на строгом разделении во времени процесса выдвижения идей и процесса их обсуждения и оценки. На первой стадии штурма запрещается осуждать выдвинутые идеи и предложения (считается, что критические замечания уводят к частностям, прерывают творческий процесс, мешают выдвижению идей). Роль аналитика состоит в том, чтобы активизировать творческое мышление участников заседания и обеспечить выдвижение возможно большего числа идей.

# Простейший алгоритм извлечения знаний из текстов включает следующие шаги.

1. Составить «базовый» список литературы для ознакомления с предметной областью.
2. Выбрать текст для извлечения знаний.
3. Беглое знакомство с текстом. Провести консультации со специалистами для определения значений незнакомых слов.
4. Сформировать первую гипотезу о макроструктуре текста.
5. Внимательно прочитать текст и выписать ключевые слова и выражения, определив тем самым «смысловые вехи».
6. Определить связи между ключевыми словами, разработать макроструктуры текста в форме графа или реферата.
7. Сформировать новое представление знаний на основании макроструктуры текста.