

Метазнания в ЭС

1. ВЫБОР ПРАВИЛ:

П1: утечка серной кислоты → использовать анион-обменник
(стоимость: дорого, источник информации: доктор Грин, степень опасности: невелика)

П2: утечка серной кислоты → использовать уксусную кислоту
(стоимость: дешево, источник информации: практикант Грун, степень опасности: велика)

П3: прежде всего использовать правило, требующее минимальных затрат

П4: прежде всего использовать правило, внесенное в БЗ специалистом

П5: прежде всего использовать правило с минимальной степенью опасности

2. ОПРАВДАНИЕ ПРАВИЛ:

П6: утечка серной кислоты → использовать известь
(оправдание: нейтрализация, образование нерастворимого и химически неактивного вещества)

П7: утечка уксусной кислоты → использовать известь
(оправдание: нейтрализация)

П8: утечка соляной кислоты → использовать известь
(оправдание: нейтрализация)

3. ОБНАРУЖЕНИЕ ОШИБОК В ПРАВИЛАХ:

ПР01: использовать известь - *нет antecedenta*

ПР02: утечка: соляная кислота → использовать известь

ПР03: соляная кислота → использовать известь - *проверить: не совпадает ли предусловие с предусловием предыдущего правила*

П9: *если некоторое правило никогда не срабатывает, проверить его предусловие*

4. СТРАТЕГИЧЕСКИЕ ПРАВИЛА:

П10: пространство поиска относительно мало → оправдан полный перебор

П11: один из конъюнктов часто ложен → перенести его в начало

П12: фрагмент часто выполняется → оптимизировать его

П13: фрагмент часто выполняется & редко меняется → скомпилировать его

П14: утечка вещества, которое не описано в БЗ → база знаний по утечкам неадекватна

Пример вывода в ЭС, основанной на правилах продукции

Правила: R1: разлита горячая жидкость → звонить по телефону 01
R2: разлита уксусная кислота → использовать известь
R3: pH жидкости < 6 → кислота
R4: кислота & имеет запах уксуса → уксусная кислота

Факты: F1: разлита жидкость
F2: pH жидкости < 6
F3: жидкость имеет запах уксуса

Разработка разума:

& { задача требует оперирования символами
 задача требует эвристических решений
 задача не слишком проста
 задача имеет практический интерес
 задача решаема (ЭС реализуема)

Трудности, Ловушки, Советы

ВЫБОР ЗАДАЧИ: задача слишком трудна, построенная ЭС не решает нужные задачи.

РЕСУРСЫ: количество разработчиков, ЭС - не обычная программа.

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА: неэффективны, не начинать с языков низкого уровня.

ЭКСПЕРТЫ: выбор Э, у Э нет времени, Э далек от компьютеров, Э незнаком с терминологией ЭС, правила слишком коротки и просты, правила слишком громоздки, Э теряет интерес к работе, слишком много Э (они не единодушны во мнениях).

РЕАЛИЗАЦИЯ: разделять знания Э и знания для решения задач, в готовой ЭС нет многих понятий, очень много специфических частных правил, нет хороших объяснений, нет удобств для пользователя, разочарование на этапе тестирования, трудно вносить изменения (большая система).

Извлечение экспертных знаний и формирование БЗ:**Эмпирические правила:**

"Чем более компетентен эксперт, тем менее способен он описать те знания, которые использует при решении задач".

"Не будьте своим собственным экспертом".

"Не принимайте на веру все, что говорят эксперты".

Методы извлечения экспертных знаний:**Наблюдение на рабочем месте**

Э решает реальные задачи, ИЗ - пассивно наблюдает

цель: ИЗ получает представление о характерных задачах.

Обсуждение задач

ИЗ обсуждает с Э отобранные им (ИЗ) характерные задачи

цель: ИЗ узнает, как организованы знания Э (понятия, гипотезы), как Э работает с неполной, неточной, противоречивой информацией, какие процедуры необходимы для решения задач.

Описание задач

ИЗ просит Э описать типичные задачи для каждого класса задач

цель: ИЗ узнает, как связаны между собой задачи одного класса, классы задач.

Анализ задач

ИЗ предлагает Э задачи и расспрашивает о ходе решения

цель: ИЗ пытается найти и сформулировать стратегии решения задач.

Доводка системы

Э предлагает ИЗ/прототипу ЭС характерные задачи

цель: ИЗ проверяет сформированную совокупность знаний (БЗ).

Оценивание системы

Э анализирует и оценивает правила, стратегии, систему понятий ПО

цель: Э оценивает точность работы ИЗ и правильность сформированной БЗ.

Проверка системы

ИЗ предлагает независимым экспертам протоколы решения задач Э и прототипом ЭС

цель: объективная оценка результатов работы ИЗ и Э (и сформированной БЗ).

Пример: Оценка размера страховых выплат

Ситуация. Беседа Инженера_знаний с Экспертом.

Некоторые правила:

Если *повреждение истца действительно требует, чтобы он носил очки*

и *истец до повреждения не носил очков*

увеличить фактор неудобства на 2500\$

Если *истец действительно имеет вероятность заболеть глаукомой*

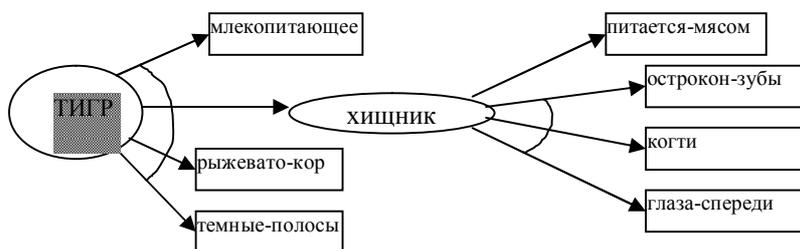
и *эта вероятность была вызвана повреждением*

и *величина этой вероятности равна 10%*

увеличить фактор осложнений в будущем на 30000\$

Реализация модельной экспертной системы на языке Плэнер

Модельная экспертная система решает задачу распознавания вида животного (*тигр*) на основе группы взаимосвязанных признаков:



Система работает с базой данных, в которой хранятся утверждения следующих типов:

(<имя> = <вид>) → ОБСН: <обоснование> ()
 (<имя> - <класс>) → ОБСН: <обоснование> ()
 (<имя> имеет <свойство>) → ОТВ: ДА НЕТ
 (<имя> питается <пища>)

Для определения вида животного используется аппарат теорем. Часть информации, необходимой для решения задачи представлена в виде утверждений базы данных, часть запрашивается у пользователя. В списках свойств утверждений представлены ответы пользователя и обоснование решения системы.

```

[define main (lambda ( )
  [prog (ОТВ ИМЯ КЛАСС ОБ)
    [while Т
      [print 'Работает эксперт по распознаванию животных']
      [print 'У Вас есть задача на распознавание?']
      [pset ОТВ [yes-no]]
      [cond ([eq .ОТВ НЕТ] [exit main main])]
      [print 'Введите имя распознаваемого объекта']
      [pset ИМЯ [read]]
      [if ([perm [goal (.ИМЯ = *КЛАСС)]]
        [pset ОБ [geta (.ИМЯ = *КЛАСС) ОБСН]]
        [output .ИМЯ .КЛАСС .ОБ] )
        (Т [mprint Кем является .ИМЯ я не знаю ])]
      [cleardb ] ])]

[define Тигр? (conseq (ИМЯ ОБ ОБ1 ОБ2)
  (*ИМЯ = ТИГР)
  [goal (.ИМЯ - млекопитающее)]
  [goal (.ИМЯ - хищник)]
  [goal (.ИМЯ имеет рыжевато-кор) (test ОТВ ДА)]
  [goal (.ИМЯ имеет темные-полосы) (test ОТВ ДА)]
  [pset ОБ1 [geta (.ИМЯ - млекопитающее) ОБСН]]
  [pset ОБ2 [geta (.ИМЯ - хищник) ОБСН]]
  [pset ОБ ( ( (.ИМЯ = ТИГР) так как (.ИМЯ – млекопитающее)
    (.ИМЯ – хищник)
    (.ИМЯ имеет рыжевато-коричневую окраску)
    (.ИМЯ имеет темные полосы)) !.ОБ1 !.ОБ2)]
  [passert (.ИМЯ = ТИГР) (with ОБСН .ОБ) ] )

[define Окраска? (conseq (ИМЯ ОТВ)
  (*ИМЯ имеет рыжевато-кор)
  [if ([search1 (.ИМЯ имеет рыжевато-кор) (test ОТВ *ОТВ)]]
    (Т [mprint .ИМЯ имеет рыжевато-коричневую окраску?]
      [pset ОТВ [yes-no]]
      [passert (.ИМЯ имеет рыжевато-кор) (with ОТВ .ОТВ)]]
    [cond ([eq .ОТВ НЕТ] [fail]]) )
  
```

```

[define Полосы? (conseq (ИМЯ ОТВ)
  (*ИМЯ имеет темные-полосы)
  [if ([search1 (.ИМЯ имеет темные-полосы) (test ОТВ *ОТВ)])
    (Т [mprint .ИМЯ имеет темные полосы?]
      [pset [yes-no]]
      [passert (.ИМЯ имеет темные-полосы ) (with ОТВ .ОТВ)])]
    [cond ([eq .ОТВ НЕТ] [fail])])])

[define Хищник1? (conseq (ИМЯ ОБ)
  (*ИМЯ - хищник)
  [goal (.ИМЯ питается мясом)]
  [pset ОБ ( ( (.ИМЯ - хищник) так как (питается мясом)))]
  [passert (.ИМЯ - хищник) (with ОБСН .ОБ) ]])

[define Хищник2? (conseq (ИМЯ ОБ)
  (*ИМЯ - хищник)
  [goal (.ИМЯ имеет острокон-зубы)]
  [goal (.ИМЯ имеет когти)]
  [goal (.ИМЯ имеет глаза-спереди)]
  [pset ОБ ( ( (.ИМЯ - хищник) так как (.ИМЯ имеет остроконечные зубы)
    (.ИМЯ имеет когти)
    (.ИМЯ имеет глаза спереди)))]
  [passert (.ИМЯ - хищник) (with ОБСН .ОБ) ]])

[define yes-no (lambda ( ) [read])]

[define output (lambda (x y z) [prog ( ) [print .x] [print.y] [print .z]])]

[assert (Муся питается мясом)]
[assert (Муся - млекопитающее)]
[assert (Муся имеет темные-полосы) (with ОТВ ДА)]
[assert (Муся имеет рыжевато-кор) (with ОТВ ДА)]

```