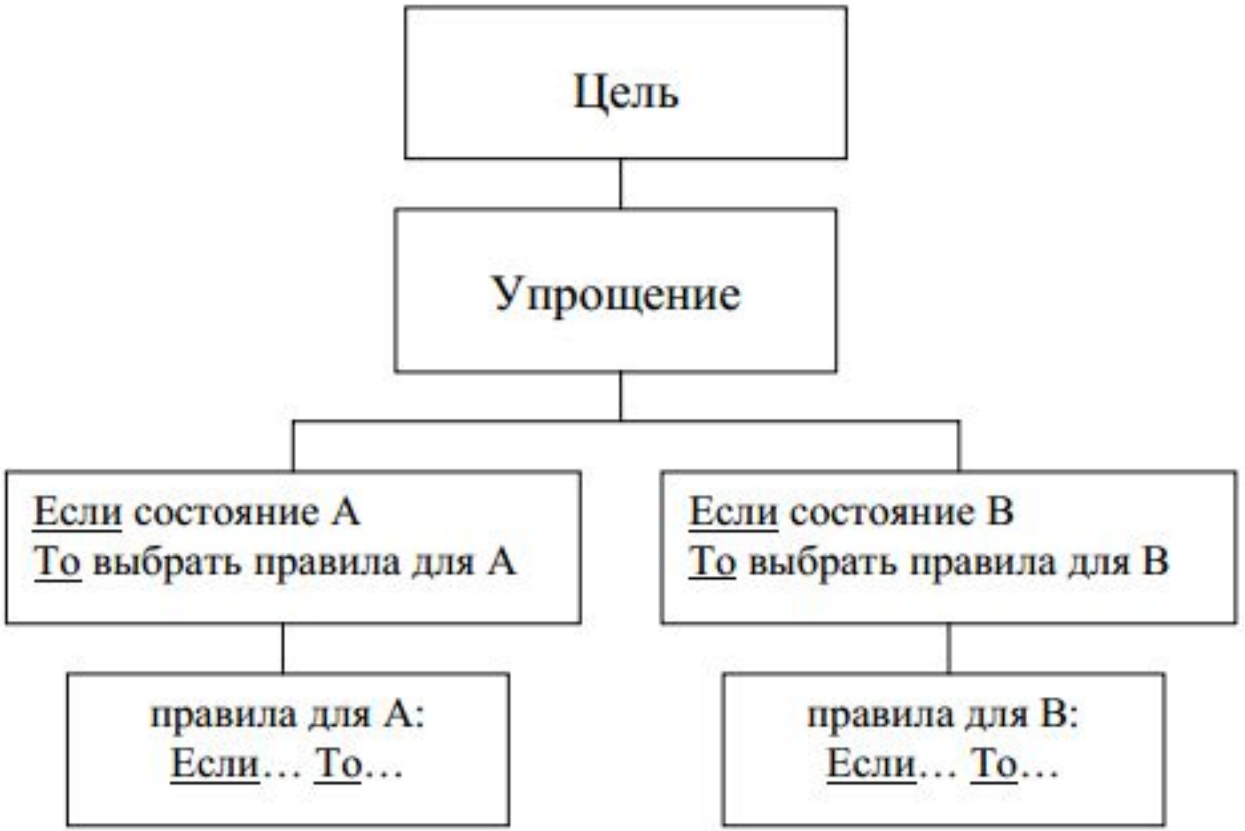
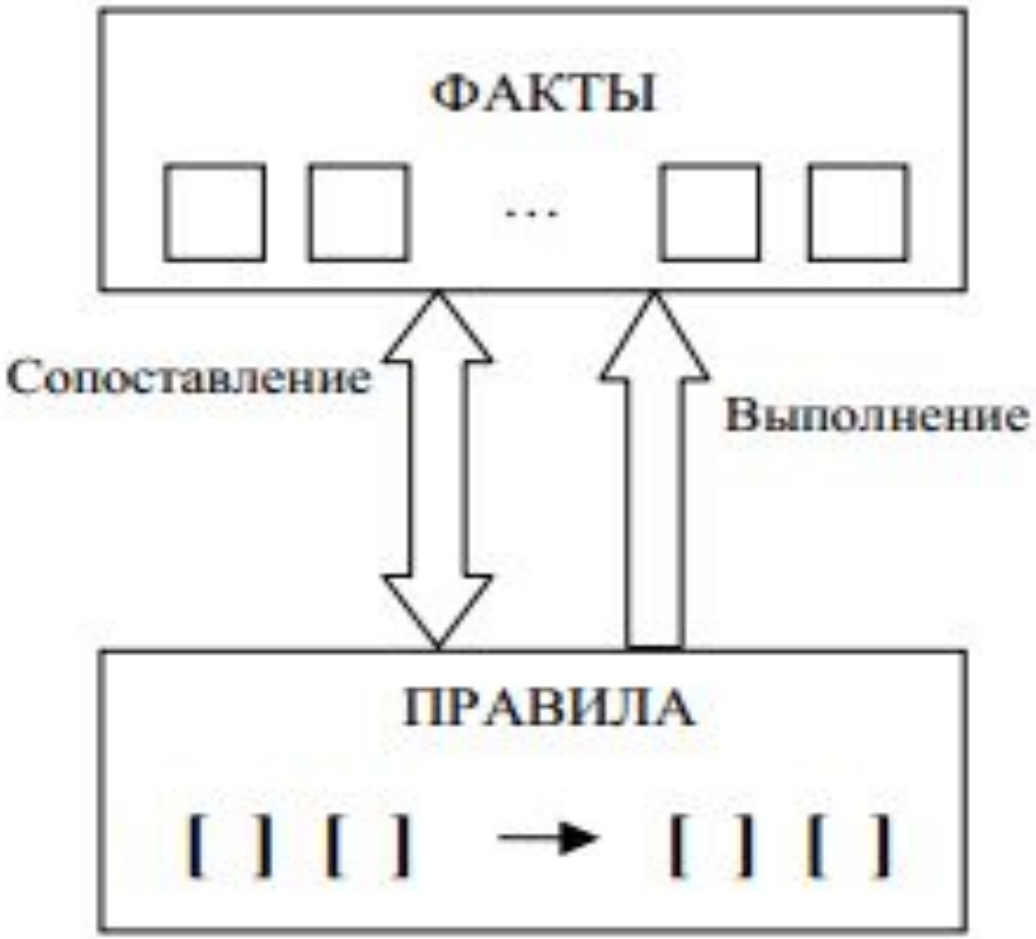
Лекция 3.
Экспертные системы

Основные составляющие процесса мышления

- Цели
- Факты и правила
- Упрощение
- Механизм вывода



Механизм вывода



Принципы построения продукционных систем

Системы обработки знаний, использующие продукционную модель получили название ***«продукционных систем».***

Продукционные правила принято записывать в виде антецедент-консеквент.

Пример продукционного правила:

ЕСЛИ
«двигатель не заводится»
и
«стартер двигателя не работает»
ТО
«неполадки в системе электропитания стартера»

Принципы построения продукционных систем

Существуют два типа продукционных систем – с «прямыми» и «обратными» выводами.

Прямые выводы реализуют стратегию «от фактов к заключениям».

При ***обратных*** выводах выдвигаются гипотезы вероятностных заключений, которые могут быть подтверждены или опровергнуты на основании фактов, поступающих в рабочую память.

Существуют также системы с ***двунаправленными выводами.***

+/- продукционных систем

Основные достоинства систем, основанных на продукционных моделях:

- Простота представления знаний
- Простота организации логического вывода

К недостаткам таких систем можно отнести следующее:

- отличие от структур знаний, свойственных человеку;
- неясность взаимных отношений правил;
- сложность оценки целостного образа знаний;
- низкая эффективность обработки знаний.

Что такое экспертная система

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 определённой целевой функции;
- не существует алгоритмического решения задачи;
- если алгоритмическое решение есть, то его нельзя использовать из-за ограниченности ресурсов (время, память).

Что такое экспертная система

Экспертная система - это программное средство, использующее экспертные знания для обеспечения высокоэффективного решения неформализованных задач в узкой предметной области. Основу ЭС составляет база знаний (БЗ) о предметной области, которая накапливается в процессе построения и эксплуатации ЭС. Накопление и организация знаний - важнейшее свойство всех ЭС.

Что такое экспертная система

Перечень типовых задач, решаемых экспертными системами, включает:

- извлечение информации из первичных данных;
- диагностика неисправностей;
- структурный анализ сложных объектов;
- выбор конфигурации сложных многокомпонентных систем;
- планирование последовательности выполнения операций, приводящих к заданной цели.

Преимущества использования ЭС

1) *Её постоянство.*

2) *Лёгкость передачи или воспроизведения.*

3) *Устойчивость* и *воспроизводимость результатов.*

4) *Стоимость.*

Отличие ЭС от традиционных программ

ЭС должны обладать:

1. Компетентностью, а именно:
- Достигать экспертного уровня решений.
- Быть умелой.
- Иметь адекватную робастность.

2. Возможностью к символьным рассуждениям, а именно:
- Представлять знания в символьном виде.
- Переформулировать символьные знания.

Отличие ЭС от традиционных программ

ЭС должны обладать:

3. Глубиной, а именно:
- Работать в предметной области, содержащей трудные задачи.
- Использовать сложные правила.

4. Самосознанием, а именно:
- Исследовать свои рассуждения.
- Объяснять свои действия.

Отличие ЭС от традиционных программ

Экспертные системы отличаются не только от традиционных программ, но и от других видов программ из области искусственного интеллекта.

- Экспертные системы имеют дело с предметами *реального мира,* операции с которыми обычно требуют наличия значительного опыта, накопленного человеком.

- Одной из основных характеристик экспертной системы является ее *производительность,* т.е. скорость получения результата и его достоверность (надежность).

Состав и взаимодействие участников построения и эксплуатации ЭС

Состав и взаимодействие участников построения и эксплуатации ЭС

- *Экспертная система* – программное средство, использующее знания экспертов для высокоэффективного решения задач в интересующей пользователя предметной области.
- *Эксперт* – человек, способный ясно выражать свои мысли и пользующийся репутацией специалиста, умеющего находить правильные решения проблем в конкретной предметной области.
- *Инженер знаний* – человек, имеющий познания в информатике и искусственном интеллекте и знающий, как надо строить ЭС.
- *Средство построения ЭС* – программное средство, используемое инженером знаний или программистом для построения ЭС.
- *Пользователь* – человек, который использует уже построенную ЭС:
  - создатель инструмента, отлаживающий средство построения ЭС;
  - инженер знаний, уточняющий существующие в ЭС знания;
  - эксперт, добавляющий в систему новые знания;
  - клерк, заносящий в систему текущую информацию.

Особенности построения и реализации ЭС

Коэффициент доверия – это число, которое означает вероятность или степень уверенности, с которой можно считать данный факт или правило достоверным или справедливым.

Система, основанная на знаниях, – это любая система, процесс работы которой основан на применении правил отношений к символическому представлению знаний, а не на использовании алгоритмических или статистических методов.

Механизм вывода содержит:

- интерпретатор, определяющий как применять правила для вывода новых знаний на основе информации, хранящейся в БЗ;
- диспетчер, устанавливающий порядок применения этих правил.

Особенности построения и реализации ЭС

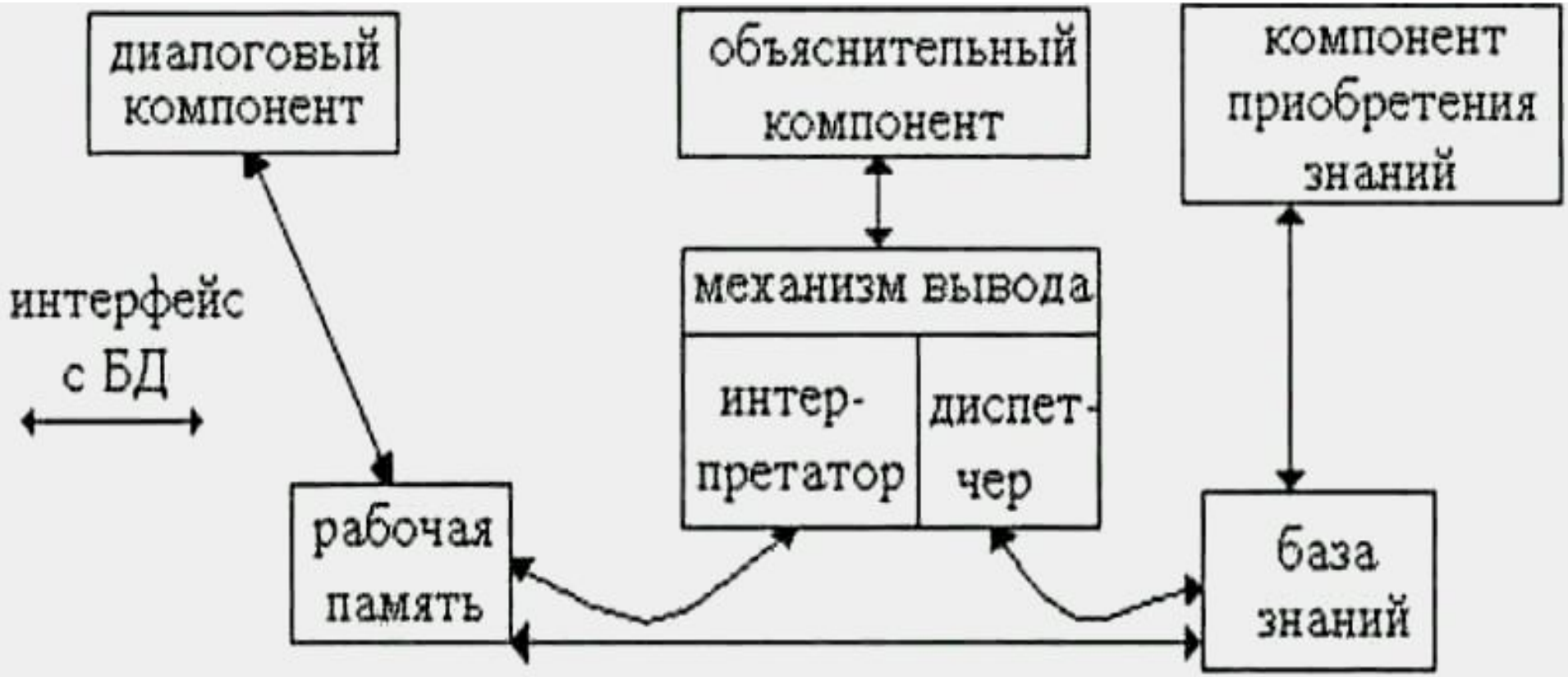


Рис.3. Структура статической ЭС

Особенности построения и реализации ЭС

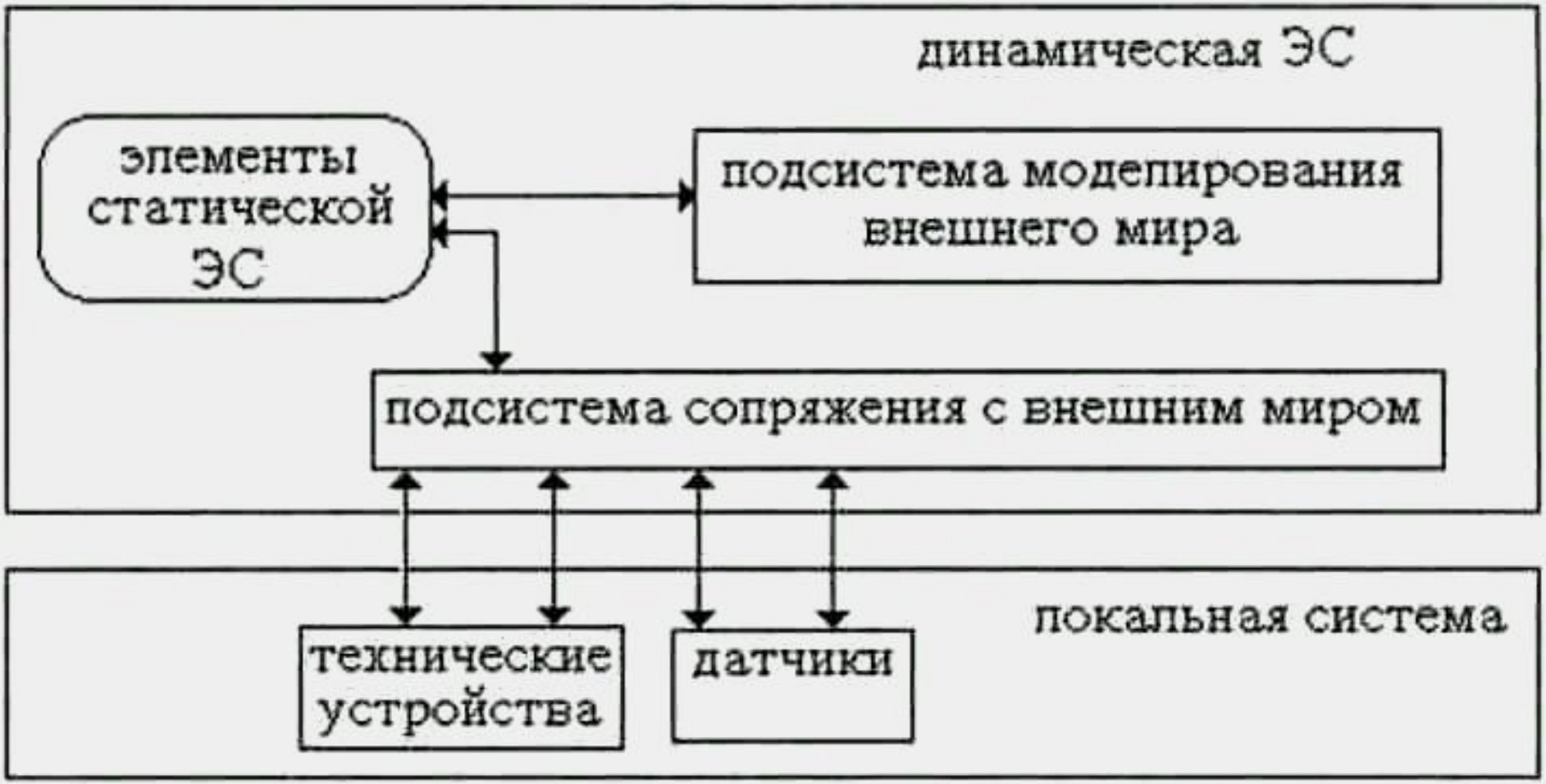


Рис.4. Динамическая экспертная система

Основные режимы работы ЭС

В работе ЭС можно выделить два основных режима: режим приобретения знаний и режим решения задачи (режим консультации или режим использования).

В режиме *консультаций* общение с ЭС осуществляет конечный пользователь. Необходимо отметить, что в зависимости от назначения ЭС пользователь может:

- не быть специалистом в данной предметной области, и в этом случае он обращается к ЭС за результатом, который не умеет получить сам;
- быть специалистом, и в этом случае он обращается к ЭС с целью ускорения получения результата, возлагая на ЭС рутинную работу.

Классификация ЭС по областям применения

- *Интерпретирующие системы*
- *Прогнозирующие системы*
- *Диагностические системы*
- *Системы проектирования*
- *Системы планирования*
- *Системы мониторинга*
- *Наладочные системы*
- *Системы оказания помощи при ремонте оборудования*
- *Обучающие системы*
- *Системы контроля*

Приобретение знаний

Приобретение знаний – это передача потенциального опыта решения проблемы от некоторого источника знаний и преобразование его в вид, который позволяет использовать эти знания в программе.

Причины низкой производительности передачи знаний:

- Специалисты в узкой области пользуются собственным жаргоном, который трудно перевести на обычный "человеческий" язык.
- Факты и принципы, лежащие в основе многих специфических областей знания эксперта, не могут быть четко сформулированы в терминах математической теории, свойства которой хорошо понятны.
- Для решения проблемы в определенной области эксперту недостаточно обладать суммой знаний о фактах и принципах в этой области.
- Экспертный анализ часто нужно поместить в довольно обширный контекст, который включает многие вещи, кажущиеся эксперту само собой разумеющимися, но для постороннего таковыми не являющиеся.

Свойства языков представления знаний

- *Логическая адекватность* означает, что представление должно обладать способностью распознавать все отличия, которые вы закладываете в исходную сущность.
- *Эвристическая мощность* означает, что наряду с наличием выразительного языка представления должно существовать некоторое средство использования представлений, сконструированных и интерпретируемых таким образом, чтобы с их помощью можно было решить проблему.
- *Естественность нотации* следует рассматривать как некую добродетель системы, поскольку большинство приложений, построенных на базе экспертных систем, нуждается в накоплении большого объема знаний, а решить такую задачу довольно трудно, если соглашения в языке представления слишком сложны.

Разъяснение принятого решения

- *Пользователи,* работающие с системой, нуждаются в подтверждении того, что заключение, к которому пришла программа, в основном корректно.
- *Инженеры, имеющие дело с формированием базы знаний,* должны убедиться, что сформулированные ими знания применены правильно, в том числе и в случае, когда существует прототип.
- *Экспертам в предметной области* желательно проследить ход рассуждений и способ использования тех сведений, которые с их слов были введены в базу знаний.
- *Программистам,* которые отлаживают и модернизируют систему, нужно иметь инструмент, позволяющий заглянуть в "ее нутро" на уровне более высоком, чем вызов отдельных языковых процедур.
- *Менеджер системы,* использующей экспертную технологию, который в конце концов несет ответственность за последствия решения, принятого программой, также нуждается в подтверждении, что эти решения достаточно обоснованы.

ЭС. Пример

В простейшем случае проблемы с двигателем могут делиться на две категории «Работает нестабильно» и «Не заводится».

Если двигатель не заводится, то он может либо совсем не раскручиваться, либо раскручиваться от аккумулятора, но не заводиться.

Если двигатель не заводится, то проблема может заключаться в отсутствии топлива или неисправности системы зажигания.

Если двигатель не раскручивается, то проблема может быть в аккумуляторной батарее или в стартере двигателя.

Если двигатель работает нестабильно, виной может быть система зажигания, а также неисправность в системе подачи топлива.

ЭС. Пример

```
(clear)
(deffunction ask (?question $?allowed)
 (printout t ?question ?allowed)
 (bind ?answer (read))
?answer )

(deffunction ask-allowed (?question $?allowed)
 (bind ?answer (ask ?question))
 (while (not (member ?answer $?allowed) )
 do
 (printout t "Reenter, please" crlf)
 (bind ?answer (ask ?question))  ) ?answer)

(deffunction ask-yes-no (?question)
 (bind ?response (ask-allowed ?question yes no))
 (eq ?response yes) )
```

ЭС. Пример

```
(defrule EngineState
 (not (work ?))
 =>
 (if (eq (ask-allowed "В чем проблема: 1) двигатель не работает 2)двигатель работает, но нестабильно" 1 2) 1) then
 (assert(work doesnot))
 else
 (assert(work unstable))  ))  // добавляем факты, определяющие состояние двигателя
```

ЭС. Пример

```
(defrule KindOfFail
 (work doesnot)
 =>
 (if (eq (ask-allowed "1 – Двигатель не крутится, 2 – Двигатель крутится, но не заводится" 1 2) 1)
 then
 (assert(engine does-not-rotate))
 else
 (assert(engine rotates))  ))
```

ЭС. Пример

```
(defrule CheckBatt   //
(engine does-not-rotate)
 =>
 (assert(suggest "Check your battery or engine starter")) )

(defrule EnoughFuel
(engine rotates)  =>
 (if (ask-yes-no "Is it enough fuel?")
 then
 (assert(need to check ignition)) else
 (assert(suggest "Add Fuel"))
)
)
```

ЭС. Пример

```
(defrule IgnitionCheck
 (not (ignition ?))
 (need to check ignition)
 =>
 (if (ask-yes-no "Check ignition system. Is there strong spark?")
 then
 (assert(ignition ok))
 else
 (assert(ignition failed))
)
)
```

ЭС. Пример

(defrule SuggestOfIgnition //если не работает зажигание и двигатель крутится – двигатель не работает из-за системы зажигания
(ignition failed)
(engine rotates)
=> (assert(suggest "Your engine does not start because of the faulty of ignition system")))

(defrule UnstableIgnition //работа двигателя не стабильна, проверяем зажигание
(work unstable)
(not (ignition ?))
=>
(assert(need to check ignition))
)

ЭС. Пример

(defrule SuggestFuel //двигатель работает нестабильно, зажигание – ок. Вывод – проблема с подачей топлива
(work unstable)
(ignition ok)
=>
(assert (suggest "Your engine work unstable due to unstable fuel supply")))

(defrule PrintSuggest //если система нашла какой-то вывод, сформировалось предположение – его выводим
(suggest ?x)
=>
(printout t ?x crlf)
)

ЭС. Пример

```
(defrule NoSuggest //если вывода нет, выводим, что система не нашла ответ
 (declare (salience -10))
 (not (suggest ?)) => (printout t "Sorry, there is no suggest." crlf))
```