

Общество с ограниченной ответственностью  
«ИНСИКОМ»  
(Интеллектуальные Системы и Комплексы)

Сетевая многопользовательская экспертная оболочка  
W.E.S.T. 1.0 (Web Expert System Toolkit)  
Оболочка - интерпретатор баз знаний

Руководство по использованию

Новосибирск 2003

<b>СОДЕРЖАНИЕ</b>	2
Введение.	3
1. Общие положения.	3
2. Установка экспертной оболочки.	4
3. Система администрирования оболочки.	5
4. Система редактирования оболочки.	7
5. Язык представления баз знаний.	11
6. Диалог с пользователем.	17
7. Дополнительные возможности.	23

**Введение.**

**W.E.S.T.** (Web Expert System Toolkit) **1.0** – модульная сетевая многопользовательская экспертная оболочка, реализованная на языке программирования PHP4, с использованием базы данных MySQL.

**1. Общие положения.**

**W.E.S.T.** — программная оболочка-интерпретатор для работы с продукционно-фреймовыми экспертными системами. Описываемая программная оболочка предназначена для решения задач обратного логического вывода на основе фреймов и правил-продукций баз знаний (интерпретации баз знаний).

Экспертные системы — это прикладные системы искусственного интеллекта. Экспертные системы предназначены для замены экспертов при решении задач в случае недостаточного количества экспертов, недостаточной оперативности в решении задачи или в опасных (вредных) для экспертов условиях. В экспертных системах базы знаний представляют собой формализованные эмпирические знания высококвалифицированных специалистов (экспертов) в какой-либо узкой предметной области.

Знания в базе знаний хранятся в виде фреймов и правил-продукций.

## 2. Установка экспертной оболочки.

Для установки экспертной оболочки W.E.S.T. 1.0 необходимо сделать следующие действия:

1. Скопировать файлы оболочки на сервер.
2. Посмотреть, чтобы параметры PHP(файл `php.ini`) были установлены в следующие значения :
  1. `file_upload = on`; //возможность передачи файлов(если есть такой параметр, если нет то нечего страшного ).
  2. `upload_max_size = 8M`; (можно и меньше, зависит от размеров баз знаний).
  3. `max_execution_time = 60`; (максимальное время исполнения программы 60сек. Можно сделать и больше, перевод баз знаний во внутреннее представление оболочки требует долгих вычисление.)
3. Отредактировать файл `include/init.php`.
  1. Прописать переменную `$host` – имя сервера (для базы данных).
  2. Прописать переменную `$login` – имя пользователя (для базы данных).
  3. Прописать переменную `$password` – пароль пользователя (для базы данных).
  4. Прописать переменную `$system_database` – имя базы данных.
5. В адресной строке броузера набрать электронный адрес оболочки и добавить `/admin`.
6. Не набирая пароля и имени пользователя, войти в систему. Система предложит произвести установку. Возможны два варианта установки:
  1. База данных и таблицы автоматически создаются сомой оболочкой.
  2. База данных создается вручную, а таблицы оболочкой.

Установить.

7. Перейти по ссылке приступить к администрированию.
8. Не набирая пароля и имени пользователя, войти в систему. Система предложит создать администратора *root* и снова зайти в систему администрирования.
9. Создать администратора *root* и снова зайти в систему администрирования. Но теперь уже будет необходимо ввести правильно пароль и логин.

Необходимость создания администратора *root* заключается в особенности системы администрирования. Каждая база знаний имеет своего хозяина, в системе администрирования есть главный администратор *root* и другие администраторы. Администратор *root* наделен всеми правами. Он может создавать и удалять администраторов, получает доступ ко всем базам знаний. Остальные же администраторы имеют доступ только к своим базам знаний и не могут добавлять новых администраторов. Такой принцип позаимствован из операционной системы UNIX, и считается приемлемым к данной экспертной оболочке.

### **3. Система администрирования оболочки.**

**Примечание:** после завершения редактирования и администрирования оболочки рекомендуется корректно выходить из системы.

На рисунке 1 показана система администрирования для администратора *root*, а на рисунке 2 система администрирования для администратора *tirex*. Как видно администратор *tirex* лишен прав добавления нового администратора .

**Добро пожаловать root в систему администрирования 1.2**

Поменять пароль:

 Пароль  
 Повторить пароль

Редактировать экспертную оболочку:

Добавить администратора:

 Login  
 Пароль  
 Повторить пароль

Login	Admin_User	Статус		
tirex	admin	<input type="button" value="Активен"/>	<input type="button" value="Удалить"/>	
qwerty	admin	<input type="button" value="Активен"/>	<input type="button" value="Удалить"/>	

[Логи](#) [Статистика](#)

Сообщения:

[Выйти](#)

Рис.1

**Добро пожаловать tirex в систему администрирования 1.2**

Поменять пароль:

Редактировать экспертную оболочку:

**Статистика**

Сообщения:

**Выйти**

Рис.2

#### 4. Система редактирования оболочки.

В системе редактирования экспертной оболочки администратор может :

- добавить новую базу знаний
- добавить существующую базу знаний в базе данных
- прикрепить файлы
- удалить прикрепленные файлы
- изменить статус базы знаний
- уничтожить базу знаний
- удалить базу знаний

*Добавить новую базу знаний* - администратор может добавить новую базу знаний. Для этого ему необходимо:

1. Выбрать текстовый файл описания базы знаний с кодировкой koï8-r, или перевести исходный файл в файл с кодировкой koï8-r.
2. Указать имя базы знаний под которым она будет храниться в базе данных.

3. Указать описание базы знаний и выбрать к какой предметной области она предназначена, если такой нет предметной области, то администратор вправе завести свою предметную область.
4. Нажать на кнопку *добавить базу знаний*, после чего последует лексико-синтаксический анализ описания базы знаний. И если анализ пройдет удачно, то оболочка предложит добавить базу знаний.

*Изменить статус базы знаний* - у базы знаний может быть два состояния (*активна, не активна*). Когда база знаний активна, то пользователь может ею пользоваться. А когда не активна пользователь может только знать, что существует база знаний в той или иной предметной области, но не может ею пользоваться. Данная функция предназначена для временного закрытия от пользователя базы знаний для их обновления или изменения.

*Удалить базу знаний* -данная функция удаляет базу знаний из списка баз знаний, но не из базы данных. Базу знаний можно будет восстановить при помощи функции *добавить существующую базу знаний в базе данных*.

*Уничтожить базу знаний* - данная функция удаляет базу знаний из базы данных.

*Прикрепить файлы* - прикрепить файлы, используемые базами знаний.

*Удалить прикрепленные файлы* - удалить прикрепленные файлы, используемые базами знаний.

На рисунке 3 показана система редактирования для *root*, а на рисунке 5 для обычного администратора. Администратор *root* имеет доступ ко всем прикрепленным файлам других администраторов.



Имя базы знаний	Владелец	Дата обновления	Статус	Предмет	Описание базы знаний		
oks	root	2002-06-24	<input type="button" value="Активен"/>	<input type="button" value="Удалить"/>	<input type="button" value="Уничтожить"/>	work	Work in home
Rand	root	2002-07-13	<input type="button" value="Активен"/>	<input type="button" value="Удалить"/>	<input type="button" value="Уничтожить"/>	other	Rand
my_oks	trix	2002-06-25	<input type="button" value="Не Активен"/>	<input type="button" value="Удалить"/>	<input type="button" value="Уничтожить"/>	sdfs	sdf
my_power	trix	2002-06-25	<input type="button" value="Активен"/>	<input type="button" value="Удалить"/>	<input type="button" value="Уничтожить"/>	sdfs	power

Добавление новой базы знаний :

Текстовый файл базы знаний

Имя базы знаний

Описание базы знаний

Предмет

Добавление существующей в базе данных базы знаний :

Имя существующей базы знаний

Описание базы знаний

Предмет

Прикрепленные файлы :

- root
- trix
- asd
- qwerty
- nick

Прикрепить файл :

Файл  
(jpg,gif,png,bmp,html,txt,php)

Сообщения:

[Вернуться к администрированию](#)  
[Выйти](#)

Рис.4

**Добро пожаловать t!rex в систему редактирования 1.1**

Имя базы знаний	Владелец	Дата обновления	Статус	Предмет	Описание базы знаний	
my_oks	t!rex	2002-06-25	Не Активен	Удалить	Уничтожить	sdfsd sdf
my_power	t!rex	2002-06-25	Активен	Удалить	Уничтожить	sdfs power

Добавление новой базы знаний :

Текстовый файл базы знаний

Имя базы знаний

Описание базы знаний

other  Предмет

Добавление существующей в базе данных базы знаний :


Имя существующей базы знаний

Описание базы знаний

other  Предмет

Прикрепленные файлы :

Prog.php

Удалить 

Прикрепить файл :

Файл  
(jpg,gif,png,bmp,html,txt,php)

Сообщения:

[Вернуться к администрированию](#)  
[Выйти](#)

Рис.5

## 5. Язык представления баз знаний.

На данном этапе разработки система уже совместима с программной оболочкой ESWin для работы с продукционно-фреймовыми экспертными системами, которая предназначена для решения задач методом обратного логического вывода на основе интерпретации правил-продукций с использованием фреймов как структур данных.

Совместимость с ESWin неполная - в оболочке W.E.S.T. 1.0:

1. Не поддерживается работа с лингвистическими переменными.
2. Базы знаний должны находиться в текстовом файле с кодировкой koï8-r (можно конечно и CP-1251, но при этом необходимо изменить параметр \$default\_charset в файле *include/init.php*).
3. Не поддерживается отношение EX(запуск программы) и добавлены арифметические операции (+ - \* /,% ), поразрядные операции (~,|,&^,<<,>>), операция объединение строк ( . ) и операция проверки значения к определенному множеству ( in, nin ).
4. Специальный фрейм-класс задающий перечень целей логического вывода (то есть обозначений задач, решаемых экспертной системой) может иметь имя «Цель» или «Mission» или «Goal»,

База знаний содержит набор фреймов и правил-продукций. Формат внешнего представления базы знаний:

```
TITLE = <название экспертной системы>
COMPANY = <название предприятия>
FRAME                                // фрейм
<описание фрейма>
ENDF
...
FRAME                                // фрейм
<описание фрейма>
ENDF
RULE                                  // правило-продукция
<описание условий правила>
```

```

DO
<описание заключений правила>
ENDR
...
RULE // правило-продукция
<описание условий правила>
DO
<описание заключений правила>
ENDR

```

Фреймы используются в базе знаний для описания объектов, событий, ситуаций, прочих понятия и взаимосвязей между ними. Фрейм – это структура данных, состоящая из слотов (полей). Формат внешнего представления фреймов:

```

FRAME (<тип фрейма>) = <имя фрейма>
PARENT: <имя фрейма-родителя>
<имя слота 1>(<тип слота>)[<вопрос слота>?]{<комментарий>}:(<значение 1>;..<значение k>)
<имя слота 1>(<тип слота>)[<вопрос слота>?]{<комментарий>}:(<значение 1>;..<значение k>)
...
<имя слота 1>(<тип слота>)[<вопрос слота>?]{<комментарий>}:(<значение 1>;..<значение k>)
ENDF

```

Среди фреймов-классов выделяется специальный фрейм-класс «Цель» или «Mission» или «Goal», задающий перечень целей логического вывода (то есть обозначений задач, решаемых экспертной системой).

*Имя фрейма, фрейма-родителя, слота* – последовательность символов (кириллические и/или латинские буквы, цифры, пробелы, знаки подчеркивания).

*Тип слота* – символьный (“symbol”), численный или лингвистическая переменная. Обязательным является описание численного типа слота (описывается зарезервированным словом «численный» или «number») и лингвистической переменной (описывается зарезервированным словом «лп» или «lingvar»). Слот без описания типа по умолчанию понимается как символьный. Описание

типа слота заключается в круглые скобки (). На данном этапе разработке лингвистические переменные не реализованы.

*Вопрос слота* – любая последовательность символов, заключенная в квадратные скобки []. Вопрос слота не является обязательным. При отсутствии вопроса будет выводиться имя слота.

*Комментарий слота* – имя текстового (\*.txt, \*.html, .htm), графического файла (.bmp,\*.png,\*.gif,\*.jpg,\*.jpeg) и скриптов (\*.php,\*.php3,\*.php4,\*.phtml), заключенного в фигурные скобки {}.

Если требуется использовать несколько файлов то необходимо их перечислить через “;”. Они будут исполняться в порядке их перечисления. К примеру:

HDD {face.gif; face.txt; door.php}:(4300; 1000; 850; 250)

Комментарий слота не является обязательным.

*Значение слота* – любая последовательность символов. Значения слота разделяются точками с запятыми. Список значений слота необязателен.

Правила-продукции описывают отношения между объектами, событиями, ситуациями и прочими понятиями. На основе отношений, задаваемых в правилах, выполняется логический вывод (решение выбранной задачи). В условиях и заключениях правил присутствуют ссылки на фреймы и их слоты. Формат внешнего представления правил:

```

RULE <номер правила>
<условие 1>
<условие 2>
...
<условие m>
DO
<заключение 1>
<заключение 2>
...
<заключение n>

```

ENDR

*Номер правила* – целое число. Начало и порядок нумерации правил произвольный, предпочтительнее правила нумеровать по порядку и начинать с единицы.

Формат записи условий и заключений одинаков и имеет следующий вид:

«отношение» («имя слота»; «значение слота») «коэффициент достоверности»

Отношения в условиях и заключениях могут быть:

EQ	=	равно;
GT	>	больше;
LT	<	меньше;
<=	=<	меньше или равно
>=	=>	больше или равно
<>	!=	!= не равно
FR		вывод фрейма-экземпляра;
GO		запуск правила;
MS		выдача сообщения.

Арифметические операции:

+		сложение
-		вычитание
*		умножение
/		деление
%		остаток от деления

Поразрядные операции:

~		Not (~0=-1)
&		And
		Or
^		Xor
<<		сдвиг на один разряд влево

>> | сдвиг на один разряд вправо

Специальные операции:

. | объединение строк

in | определения принадлежности к множеству значений.

notin | определения не принадлежности к множеству значений.  
(обратная функция in)

any | опрос пользователя при любом значении слота.

Используется для получения нового факта.

В заключениях правил используются только отношения EQ, GO, FR, MS и арифметические операции +, -, \*, /, %. Специальная операция “.”.

*Имя слота* – содержит имя фрейма и имя слота, разделенных точкой.

*Значение слота* – строка или число (определяется типом слота), или имя слота. К примеру:

Rule 20

```
> (BU_Cost.Сумма;1)
!= (BU_Cost.Сумма;13)
EQ (BU_Cost.Сумма;12)
IN (BU_Cost.Сумма;12..14)
DO
GO (BU_Cost.Сумма; 17)
EQ (BU_Cost.Итого; BU_Cost.Сумма)
+ (BU_Cost.Итого; 120)
+ (BU_Cost.Итого; BU_Cost.Сумма)
. (BU_Cost.Сумма;Hello)
```

EndR

Это правило эквивалентно следующей записи на гипотетическом языке программирования (подобные выражения не поддерживаются оболочкой) :

```
If (BU_Cost.Сумма > 1) AND (BU_Cost.Сумма != 13) AND (BU_Cost.Сумма =12) AND
  (BU_Cost.Сумма IN 12..14)
THEN
  BU_Cost.Сумма GO 17
```

```

BU_Cost.Итого = BU_Cost.Сумма
BU_Cost.Итого = BU_Cost.Итого + 120
BU_Cost.Итого =BU_Cost.Итого + BU_Cost.Сумма
BU_Cost.Сумма = BU_Cost.Сумма . Hello)

```

При использовании отношения “in” или “nin” значения слота можно задать несколькими способами:

```

In(BU_Cost.Итого;100..350)
In(BU_Cost.Итого; 100;129;134..350;354)
in(Компьютер.Video;Компьютер1.Video)

```

Так же существует дополнительная возможность, использования арифметических выражений в качестве значения слота. При этом вначале выражения следует поставить специальный символ “#” В этих выражения могут использоваться все ранее перечисленные арифметические и поразрядные операции с соблюдением их приоритета и порядком выполнения с помощью скобок “()”. Специальные операции такие как, объединение строк “.” и определения принадлежности к множеству значений “in”, “nin” и “any” не могут использоваться в подобных выражения. Примеры:

```

Frame= Компьютер
A(number):()
B(number):()
Oper:()
C():()
EndF
Rule 1
  ANY(Компьютер.A;)
  ANY(Компьютер.B;)
  ANY(Компьютер.Oper;)
  <>(0;#Компьютер.B+ Компьютер.A) //арифметические выражения можно использовать и
                                  в условиях
Do
EQ(Компьютер.sum; #(Компьютер.A+ Компьютер.B)*100)//правильное выражение
.(Компьютер.C;Компьютер.A)//объединение Компьютер.C с Компьютер.A
.(Компьютер.C;Компьютер.Oper)// Компьютер.C = 'Компьютер.A Компьютер.oper'
.(Компьютер.C;Компьютер.B)// Компьютер.C='Компьютер.AКомпьютер.oper Компьютер.B'

```



```

EQ(Test;Тест пройден успешно)
EndR
Rule 2
  ANY(Компьютер.А;)
  ANY(Компьютер.В;)
  ANY(Компьютер.Орег;)
Do
EQ(Компьютер.sum;(Компьютер.А+ Компьютер.В)*100)//будет приниматься за строчку
EndR

```

*Коэффициент достоверности* – число от 0 до 100. Коэффициент достоверности в заключении используется при формировании значения слота фрейма-экземпляра при срабатывании правила. Коэффициент достоверности в условии в этой версии не используется. По умолчанию коэффициент достоверности принимает значение 100.

#### **6. Диалог с пользователем.**

На главной странице экспертной оболочки пользователю предоставляется возможность выбрать существующую базу знаний в интересующей его предметной области. Так же пользователю предоставляется возможность использовать ряд предусмотренных разработчиком функций. Вот их перечень:

- Выбор цели.
- Поиск решения.
- Следующее решение.
- База фактов.
- Трассировка.
- Очистить БФ.
- Прервать.
- Закончить работу.

*Выбор цели* - после того как пользователь выберет интересующую его предметную область и базу знаний в этой области, пользователю предоставляется задание цели логического вывода.

*Поиск решения* - поиск первого подходящего решения для заданной цели логического вывода.

*Следующее решение* - последовательный поиск следующих подходящих решений для заданной цели логического вывода.

*База фактов* - просмотр базы фактов, полученных в результате логического вывода.

*Трассировка* - просмотр последовательности действий, выполненных программной оболочкой в процессе поиска решения.

*Очистить БФ* - очистка базы фактов для очередного поиска подходящего решения для заданной цели логического вывода.

*Прервать* - прервать процесс логического вывода.

*Закончить работу* - закончить работу с экспертной оболочкой, при выполнении этой операции очищается база фактов и удаляется вся служебная информация о работе пользователя с экспертной оболочкой.

**Примечание** : функция закончить работу является обязательной для выполнения каждым пользователем, чтобы пользователь в дальнейшем мог корректно работать с оболочкой.

Общение экспертной системы с пользователем происходит с помощью диалога. В ходе которого экспертная система задает пользователю ряд наводящих вопросов и варианты ответов на них или пользователь самостоятельно формулирует ответы. В ходе полученных данных от пользователя экспертная система путем обратного логического вывода предоставляет пользователю один из возможных вариантов решения поставленной задачи в определенной предметной области.

На рисунках 6-10 представлены примеры общения экспертной системы с пользователем.

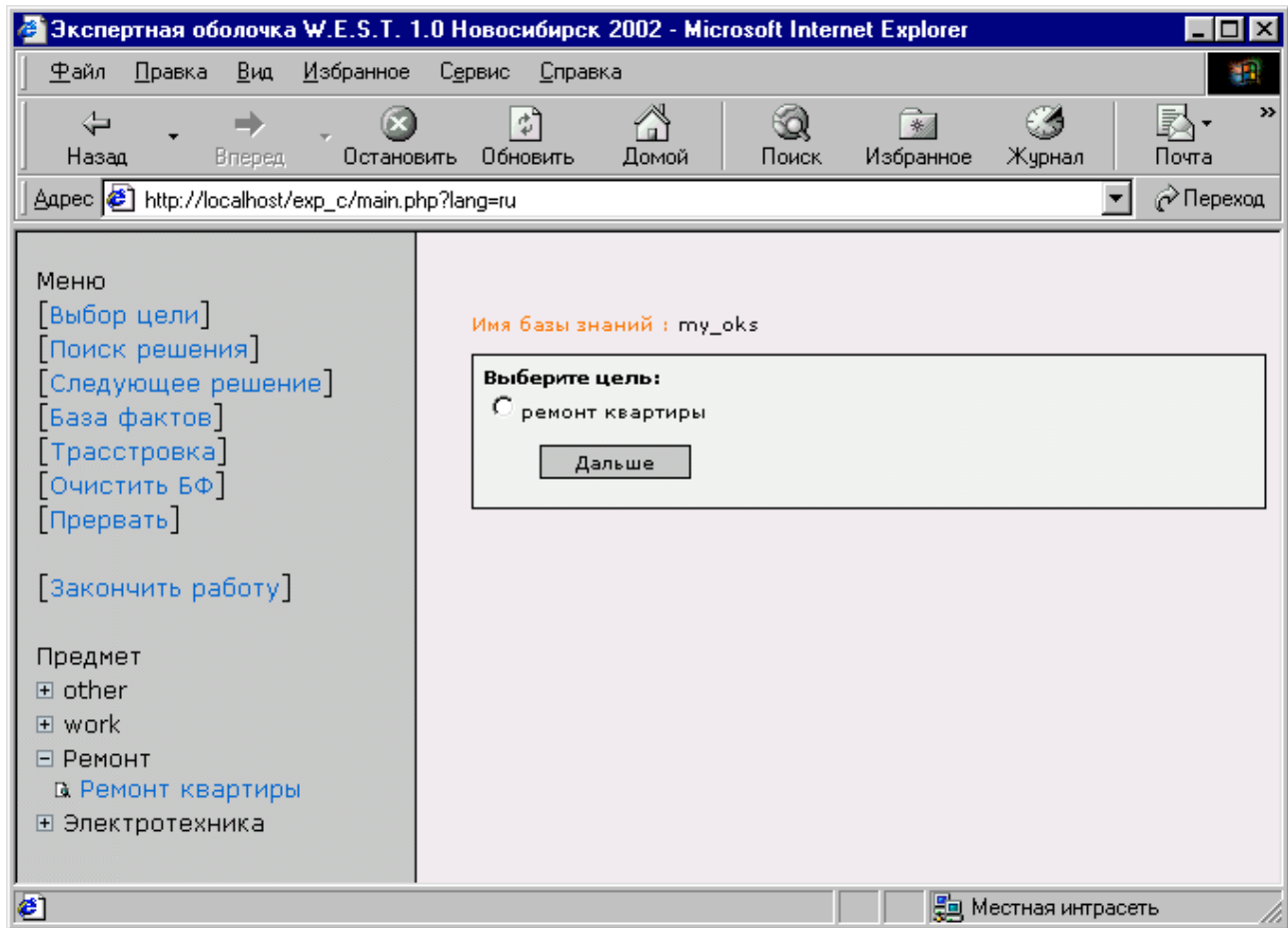


Рис.6 Выбор цели.

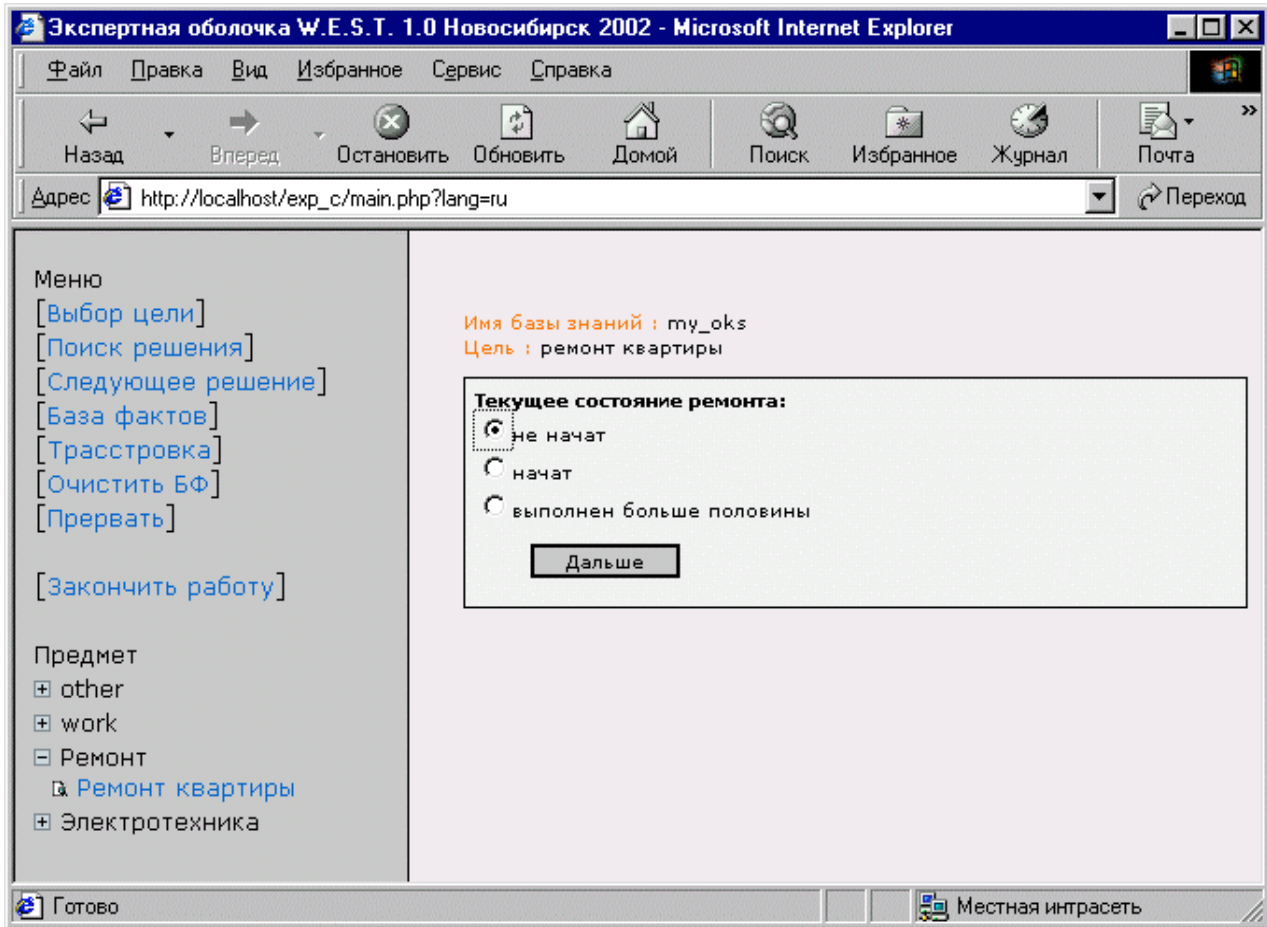


Рис.7 Ответ на наводящий вопрос.

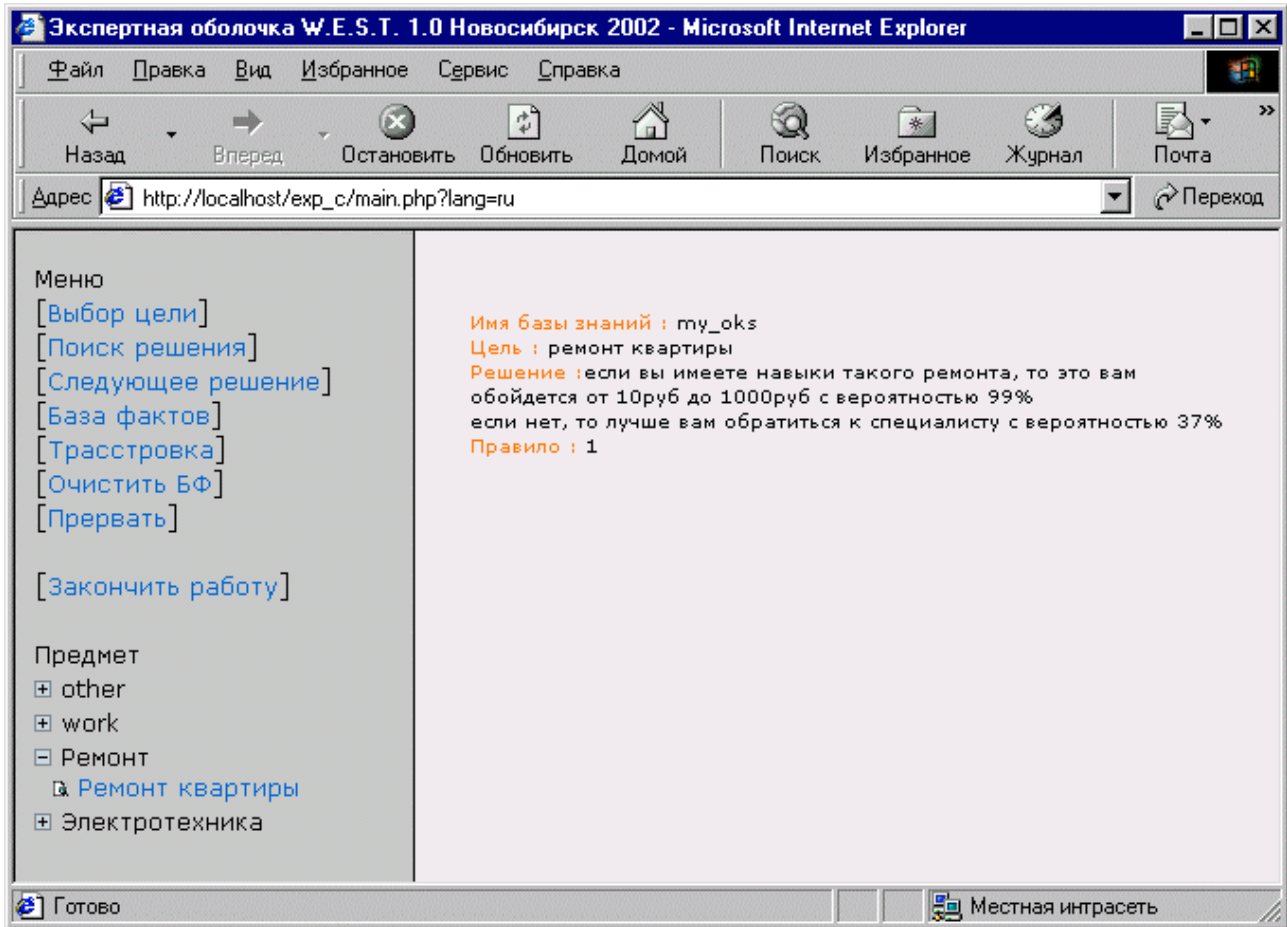


Рис.8 Вывод решения.

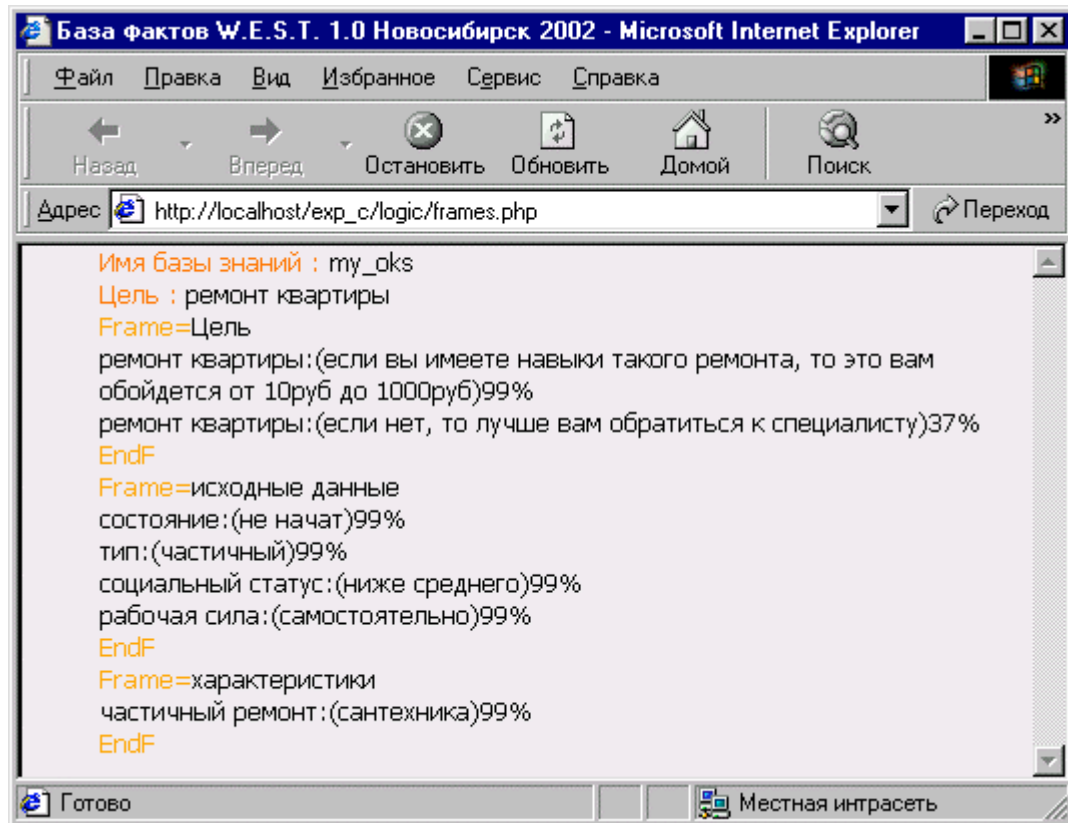


Рис.9 Просмотр базы фактов.

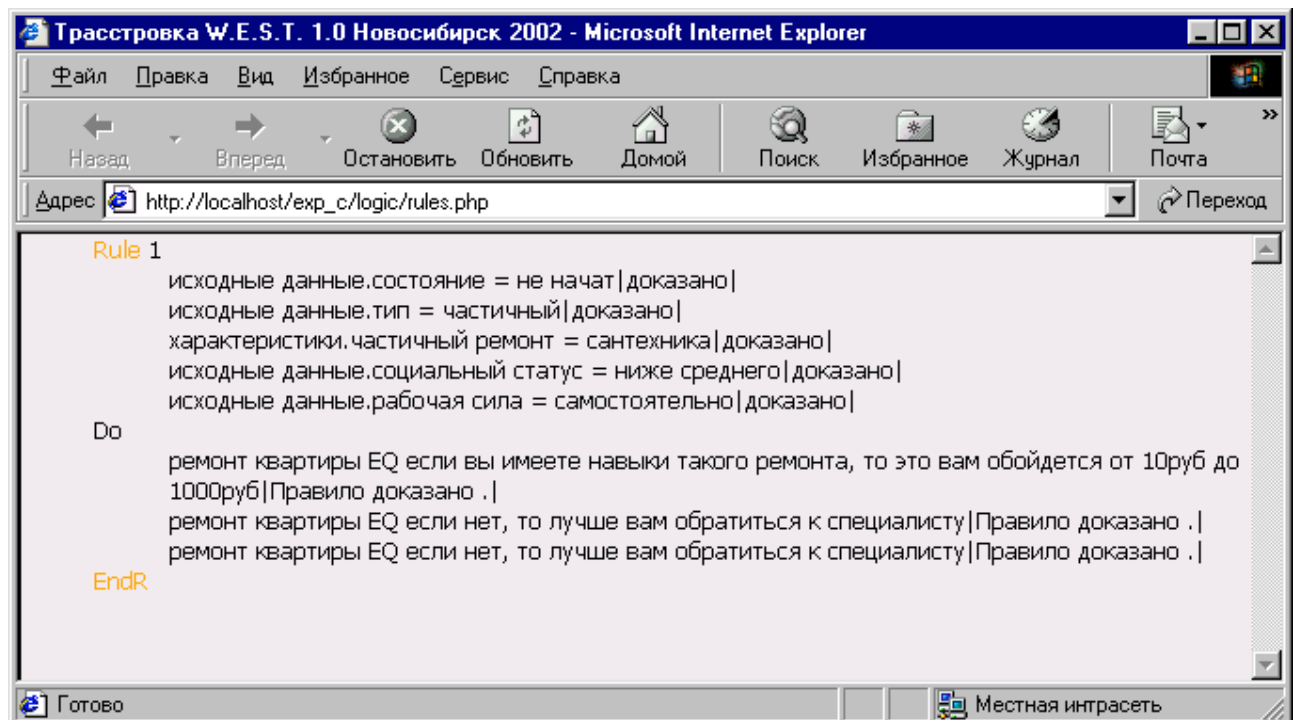


Рис.10 Просмотр трассировки.

**7. Дополнительные возможности.**

1. Многоязычный пользовательский интерфейс (Английский, Русский и т.д.)
2. Статистика посещений баз знаний.
3. Система контроля администрирования.