**Разработка (СОЗ) системы, основанной на знаниях.**

(Подготовка к разработке ЭС.)

Предварительно надо подготовиться к лабораторной работе №3, т.е. изучить АОС дихотомии, реализованную в среде Delphi. Изучить глубинные знания, представленные в файле PREDOBL.PAS в папке PrObl. Изучить поверхностные знания, представленные на языке эксперта в файле TXT1.TXT. Разобраться со связью поверхностных знаний с глубинными с помощью конструкции «параметры» языка эксперта. Надо подчеркнуть, что решение задачи дихотомии, реализовано в виде автоматизированной обучающей системы, т.е. оно проверяет ответы пользователя. В разрабатываемой программе проверку ответов пользователя можно не реализовывать.

Предполагаем, что пользователь знаком с использованием ЭС. Знает, как работают правила продукций и может сам написать их. Знает, как связаны поверхностные знания (на языке эксперта) и глубинные знания (на процедурах, написанных на языке программирования).

Из-за невозможности использования глубинных знаний инструментальной ЭС, реализованной в среде программирования Delphi, возникает необходимость разработки упрощенной СОЗ, облегчающей переход к созданию ЭС.

Цель работы: решить задачу из предметной области с интеллектуальной поддержкой.

Примеры задач:

- классификация чего-либо,

--классификация плохих блоков аппаратуры,

- диагностика,

- поиск неисправностей,

- мониторинг (слежение за больным).

- другие примеры.

Два окна: окно «Предметной области» и окно «Диалог». Хотелось бы видеть окно предметной области, в котором что-то меняется и проверяется с помощью процедур. На основании текущего состояния разработанная программа (эксперт со своим окном диалога) принимает решение. Пример – дихотомия.

Инструментарий для решения задач предметной области с помощью ЭС реализован в среде Delphi, использование которой в настоящее время не поддерживается. Но учитывая, что студенты владеют знаниями языков программирования, можем решить задачи из различных предметных областей с помощью упрощенной системы, основанной на знаниях (СОЗ).

Чтобы пользователь не заметил подмены (замены ЭС на упрощенную СОЗ) в решении его задачи, программист должен снабдить программу кучей пояснений (каждого программного действия) на русском языке. Надо иметь возможность отключать демонстрацию этих пояснений, если пользователь уже все понял. Вытекает требование реализации двух режимов: пошагового и автоматического. При автоматическом режиме пояснения не показывается и происходит обычное выполнение программы.

База знаний состоит из поверхностных знаний (ПЗ) и глубинных знаний (ГЗ). ГЗ – это процедуры предметной области. ПЗ записываются на языке эксперта, близком к естественному.

Каждый студент выбирает некую (любую) предметную область и некоторую задачу в ней. Реализует необходимые процедуры на каком-то языке программирования. Записывает решение задачи (поверхностные знания), состоящее из обращений к процедурам глубинных знания и сообщений пользователю, с различными пояснениями. Перед каждым вызовом, каждой процедуры вставляем печать сообщения, с пояснениями для данной процедуры. Для реализации сообщений используются специальные процедуры. Решение задачи записывается на том же языке программирования, что и процедуры, только на «более высоком» уровне (не пишем процедуры, а только их используем). Когда запускаем эту программу, пользователь видит сообщения на русском языке. Получается, что нашу программу понимает и машина, и человек (не программист).

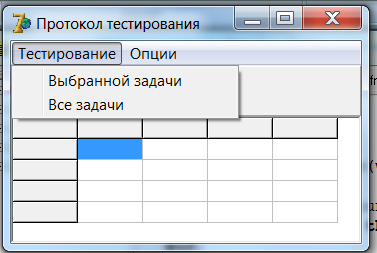
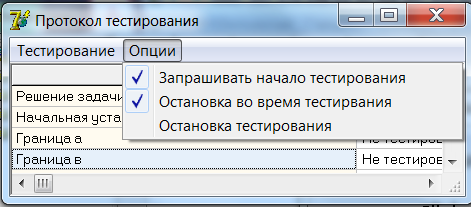
Полученные поверхностные знания легко могут быть представлены с помощью языка эксперта. Поэтому их можно назвать квазиповерхностными знаниями.

Использовать разработанную программу в двух режимах. В демонстрационном режиме – при котором показываются сообщения. В автоматическом режиме – сообщения не показываются, т.е. подтверждается работоспособность программы.

В качестве примера рассмотрим задачу дихотомии, которую мы изучили в прошлом семестре.

СОЗ для задачи дихотомии, реализованная в среде Delphi, приведена в папке «AKP». Разработанная система, позволяет использовать несколько примеров, выделенных при решении задачи дихотомии.

Результаты помещаются в следующую сетку

**Создание глубинных знаний (ГЗ).** В среде Delphi.

Создание модуля предметной области PREDOBL.PAS. Необходимо создать окно предметной области и разработать процедуры (функции), которые будут составлять глубинные знания. Для нашего примера с задачей дихотомии такими процедурами будут

*procedure initial(var a,b:real); {начальная установка}*

*procedure aa1(var x1,x2: real); {дихотомия}*

*procedure a2; {обновленная информация}*

*procedure a3(var a:real); {граница\_a}*

*procedure a4(var b:real); {граница\_b}*

*procedure a5(a,b:real; var ans2: string40); {конец}*

*procedure a6; {результаты}*

Следует подчеркнуть, что процедуры упрощены, по сравнению с реализацией АОС дихотомии на ЭС, за счет отсутствия проверки ответов пользователя.

**Создание поверхностных знаний (ПЗ).**

Файл frmtest.pas содержит базу знаний, помеченную специальными комментариями, и исполняющую систему, которая включает в себя две процедуры сообщений *testmessage и testmessage2,* которые показываются в демонстрационном режиме и не показываются в автоматическом режиме.

В текст программы (файл frmtest.pas) вставляются сообщения на русском языке, содержащие пояснения о работе процедур. То есть осуществляется интеллектуальная поддержка.

{========НАЧАЛО БАЗЫ ЗНАНИЙ===========}

{=============4\_1\_1======================}

*function Test\_1:Integer; var ans2:string40; label 1;*

*begin Result := trGood;*

*try*

*testmessage('Нажмите ОК для запуска ' + 'Дихотомии');*

*testmessage('Будет проведена начальная установка ');*

*initial(a,b); {начальная установка} testmessage('Будет проведена дихотомия ');*

*aa1(x1,x2); testmessage('Показ обновленной информации '); a2;*

*1: testmessage(' Определение границы\_a.'+#10#13+*

*' Задайте значение a (real). '+#10#13+*

*' Подсказка граница\_а ::: if f(x1)>f(x2) then a:=x1');*

*a3(a); testmessage('Показ обновленной информации '); a2;*

*testmessage('Определение границы\_b. '+#10#13+*

*'Задайте значение b (real)'+#10#13+*

*' Подсказка граница\_б ::: if f(x1)<f(x2) then b:=x2');*

*a4(b); testmessage('Показ обновленной информации '); a2;*

*testmessage('Будет проведена ДИХОТОМИЯ ');*

*aa1(x1,x2); testmessage('Показ обновленной информации '); a2;*

*testmessage('Проверка окончания'+#10#13+*

*' Точность достигнута? (да/нет) '+#10#13+*

*'подсказка конец ::: if abs(a-b)<= 2\*eps+eps/100 then да else нет');*

*if TestMessage2('Если точность достигнута,'+#10#13+*

*' то нажмите "да" иначе "нет".') = ID\_YES then*

*Result := trGood else if Not(autotest) Then Result := trExpert; a5(a,b,ans2);*

*if (ans2 <> 'да') then begin testmessage('точность не достигнута ');goto 1; end;*

*testmessage('Конец решения. '); a6;*

*except Result := trRuntime; end; end;*

**Фрагмент этой программы:**

*testmessage('Будет проведена ДИХОТОМИЯ ');*

*aa1(x1,x2); {дихотомия}*

*testmessage('Показ обновленной информации ');*

*a2; {информация}*

*1: testmessage(' Определение границы\_a.'+#10#13+*

*' Задайте значение a (real). '+#10#13+*

*' Подсказка граница\_а ::: if f(x1)>f(x2) then a:=x1');*

*a3(a); {граница\_a}*

*testmessage('Показ обновленной информации ');*

*a2; {информация}*

*testmessage('Определение границы\_b. '+#10#13+*

*'Задайте значение b (real)'+#10#13+*

*' Подсказка граница\_б ::: if f(x1)<f(x2) then b:=x2');*

*a4(b); {граница\_b}*

*testmessage('Показ обновленной информации ');*

*a2; {информация}*

**может быть представлен правилом продукции:**

*2 если дихотомия – проведена и информация - выведена*

*и граница\_а – да и граница\_б – да то шаг – сделан*

**а также параметрами, вопросами и подсказками:**

*параметры дихотомия<ВК>4 a1 word no*

*параметры информация<ВК>5 a2 word no*

*параметры обнавленная\_информация<ВК>5 a2 word no*

*параметры граница\_а<ВК>6 a3 real no*

*вопрос граница\_а<ВК>Задайте значение a (real)*

*подсказка граница\_а<ВК>::: if f(x1)>f(x2) then a:=x1*

*параметры граница\_б<ВК>7 a4 real no*

*вопрос граница\_б<ВК>Задайте значение b (real)*

*подсказка граница\_б<ВК>::: if f(x1)<f(x2) then b:=x2*

После этого, легко установить соотношение между этой программой и записью решения задачи дихотомии на языке эксперта. Связь между поверхностными знаниями и глубинными знаниями осуществляется с помощью конструкции «параметры». Таким образом, разработанная СОЗ облегчает переход к ЭС.

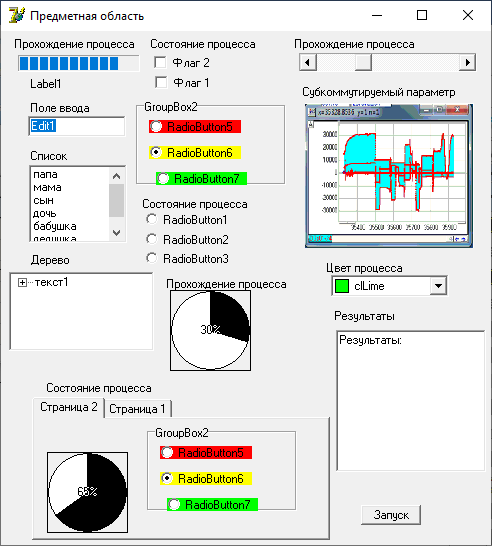
В папке «AKP\_Python» приведен пример решения задачи дихотомии на Python, с помощью, упрощенной (без интерфейса) СОЗ.

По аналогии с этим примером, каждый член бригады создает свою программу или часть общей программы на знакомом ему языке программирования.

**Варианты предметных областей и задач.**

**1.Нарисовать граф ботанической базы знаний и построить СОЗ с использованием процедур.**

**2.Используем различные графические компоненты, анализируем их состояния и принимаем решение.**



Макет

Сделать два окна. С помощью одного окна управлять вторым (окном предметной области).

**Примеры правил продукции.**

Обратить внимание на то, что параметр – одно слово (лексема). Поэтому зеленая\_лампочка.

*Если зеленая\_лампочка - горит и флаг - установлен то работа – правильная.*

*Если красная\_лампочка - горит и флаг – не\_установлен*

*то работа – неправильная и ошибка - 225.*

*Если желтая\_лампочка - горит то на\_страницу 2 - перейти*

*Если протекание\_процесса – меньше\_40 и зеленая\_лампочка – горит то ошибка -12.*

*Если флаг установлен и параметр – простой то работа – нормальная*

*Если протекание\_процесса – меньше\_40 и зеленая\_лампочка – горит то ошибка -12*

*Если в\_списке – меньше\_20 то процесс – нормальный*

**Использовать таймер для оживления процесса – получим задачу мониторинга.**

Делаем два окна. 1. Окно «ПрОбл». 2. Окно «Вопросы эксперта» - диалог.

Пользователь этой системы смотрит на окно «ПрОбл» и отвечает на вопросы эксперта.

Вопросы ПЗ:

1. Горит ли флажок? – да, нет.
2. Какая радиокнопка горит? – зеленая, желтая, красная.
3. Каково значение указателя? – 60.
4. На какой странице открыт блокнот? – 2.

Более простые варианты сообщений.

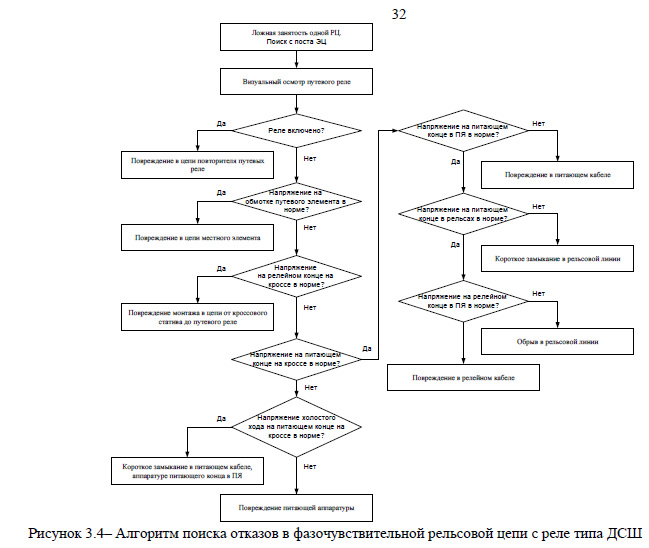
Сообщения ГЗ. Пользователю сообщают, что

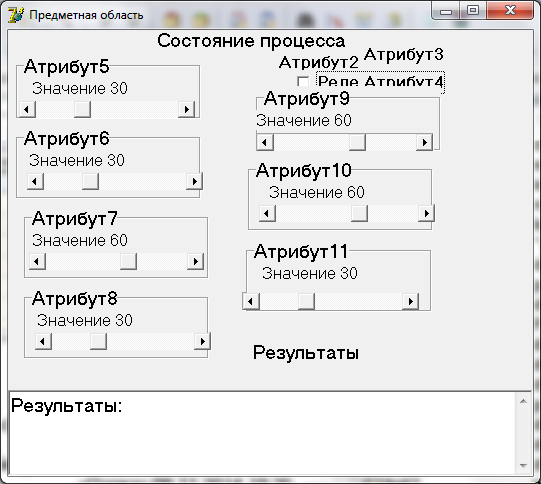
1. Флажок горит.
2. Желтая кнопка горит. (кнопка\_желтая – лексема)
3. Значение указателя – 60.
4. Блокнот открыт на 2-й странице.

**3.Поиск неисправностей на ж/д транспорте.**

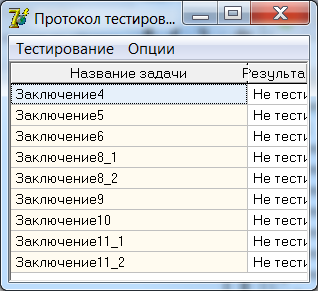
Сборник методик и алгоритмов поиска и устранения неисправностей в устройствах СЦБ

<http://scbiinfrastruktura.ru/wp-content/uploads/Sbornik-metodik-i-algoritmov-poiska-i-ustraneniya-neispravnostey-v-ustroystvah-STSB.pdf>





СОЗ



## Таблица названий атрибута и вопросов.

Название должно образовывать одну лексему.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Название атрибута | Вопрос |
| 3 | Атрибут3 | Реле | Реле включено? |
| 7 | Атрибут10 | Напряжение\_на\_релейном\_конце\_на\_кроссе | Напряжение на релейном конце на кроссе в норме? |
| 10 | Атрибут17 | Напряжение\_на\_релейном\_конце\_в\_ПЯ | Напряжение на релейном конце в ПЯ в норме? |

## Тексты заключений.

|  |  |
| --- | --- |
| Название атрибута | Текст заключения |
| заключение1 | Повреждение в цепи повторителя путевых реле |
| Заключение9 | Повреждение в релейном кабеле |

## ВОПРОСЫ

*вопрос атрибут3<ВК>Реле включено?*

*вопрос атрибут5<ВК>Напряжение на питающем конце в ПЯ в норме?*

*вопрос атрибут7<ВК>Напряжение на обмотке путевого элемента в норме?*

*вопрос атрибут8<ВК>Напряжение на питающем конце в рельсах в норме?*

*вопрос атрибут10<ВК>Напряжение на релейном конце на кроссе в норме?*

*вопрос атрибут12<ВК>Напряжение на питающем конце на кроссе в норме?*

*вопрос атрибут13<ВК>Напряжение холостого хода на питающем конце на кроссе в норме?*

*вопрос атрибут17<ВК>Напряжение на релейном конце в ПЯ в норме?*

**Глубинные знания**.

Примеры глубинных знаний.

*procedure ask1(var val : string40); //Проверка флажка*

*begin*

*if FormPredObl.CheckBox4.Checked then val:= 'да' else val:='нет';*

*ShowMessage('Реле включено? VAL='+VAL);*

*end;*

*procedure a2(var val : string40); {Проверка радиокнопки}*

*begin*

*{ if FormPredObl.RadioButton1.Checked then val:= 'да' else val:='нет'; }*

*// ShowMessage('VAL='+VAL);*

*end;*

*procedure ask5(var val : string40); //*

*begin*

*if FormPredObl.ScrollBar2.Position > 50 then val:= 'да' else val:='нет';*

*ShowMessage('Напряжение на обмотке путевого элемента в норме? VAL='+VAL);*

*end;*

*procedure result1(); {ответ}*

*begin*

*FormPredObl.Memo1.Lines.Add('Повреждение в цепи повторителя путевых реле');*

*end;*

*procedure result9(); {ответ}*

*begin*

*FormPredObl.Memo1.Lines.Add('Повреждение в релейном кабеле');*

*end;*

**Поверхностные знания.**

На первом этапе с атрибутами надо связать вопросы. При отсутствии процедуры, связанной с атрибутом, на вопрос будет отвечать пользователь. Глядя на блок-схему, пользователь сможет с помощью своих ответов, пройти по всем веткам алгоритма и проверить его правильность. Если с атрибутом связать процедуру, то для того чтобы пройти по всем веткам алгоритма, потребуется менять состояние предметной области, что значительно сложнее ответов пользователя.