

Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Санкт-Петербургский государственный университет
информационных технологий механики и оптики

Муромцев Дмитрий Ильич

**ОБОЛОЧКА ЭКСПЕРТНЫХ СИСТЕМ
EXSYS CORVID**

МЕТОДИЧЕСКОЕ ПОСОБИЕ



Санкт-Петербург 2006

УДК [004.891 + 002.53:004.89] (075.8)

Д.И. Муромцев. Оболочка экспертных систем Exsys Corvid. – СПб:
СПб ГУ ИТМО, 2006. – 69 с.

В методическом пособии представлены шесть лабораторных работ, позволяющих студентам овладеть основными навыками разработки экспертных систем в Exsys corvid v. 3. В данных работах рассматриваются основные аспекты проектирования экспертных систем, включая представление знаний в виде объектно-структурированных структур, прямой и обратный вывод. Также рассматриваются вопросы разработки графического интерфейса пользователя на базе шаблонов web-страниц.

Одобрено на заседании совета факультета компьютерных технологий управления Санкт-Петербургского государственного университета информационных технологий механики и оптики, протокол № 10 от 27.16.2006.

© к.т.н. Муромцев Д.И., 2006
© Санкт-Петербургский Государственный Университет
информационных технологий, механики и оптики, 2006

ОГЛАВЛЕНИЕ

Лабораторная работа 1: Создание простейшей системы	4
Лабораторная работа 2: Улучшение интерфейса пользователя	23
Лабораторная работа 3: Усиление логики работы системы	29
Лабораторная работа 4: Обратная связь	42
Лабораторная работа 5: Числовые переменные и [[]] подстановки.....	48
Лабораторная работа 6: Переменные коллекции.....	57
История кафедры ПКС	63

Лабораторная работа 1: Создание простейшей системы

Цель работы: изучение интерфейса Exsys CORVID на примере простейшей экспертной системы.

ЭС на базе оболочки Exsys CORVID используют правила вида IF/THEN.

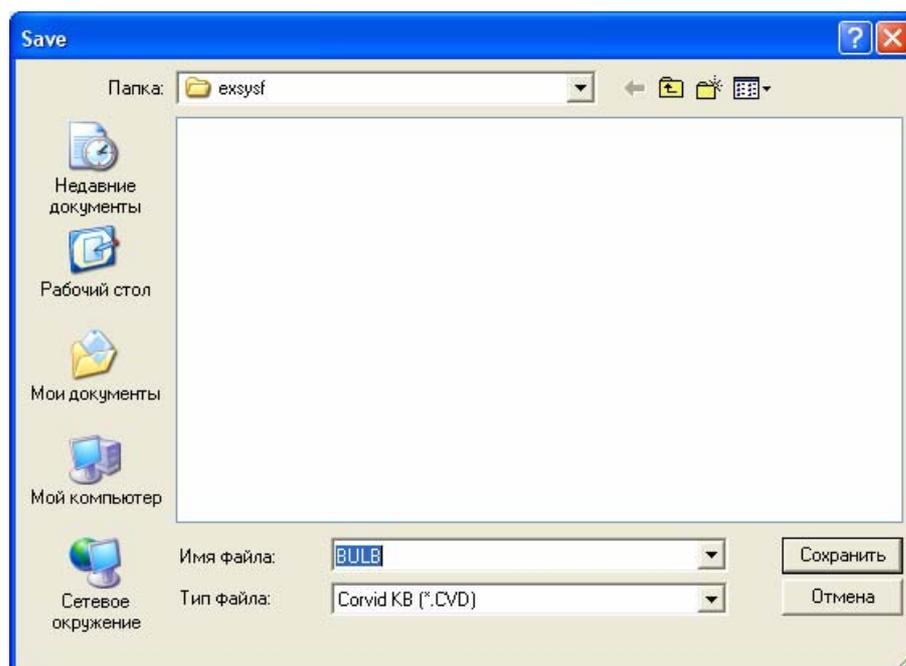
Рассмотрим простейшую ситуацию: Предположим, свет в Вашем доме внезапно погас. Возможно, Вы решите поменять лампочку. В самом простейшем случае можно написать только 2 правила:

```
IF
    Свет в Вашем доме внезапно перестал работать
THEN
    замените лампочку

IF
    Свет в Вашем доме продолжает работать
THEN
    Ничего не делать
```

Шаг 1

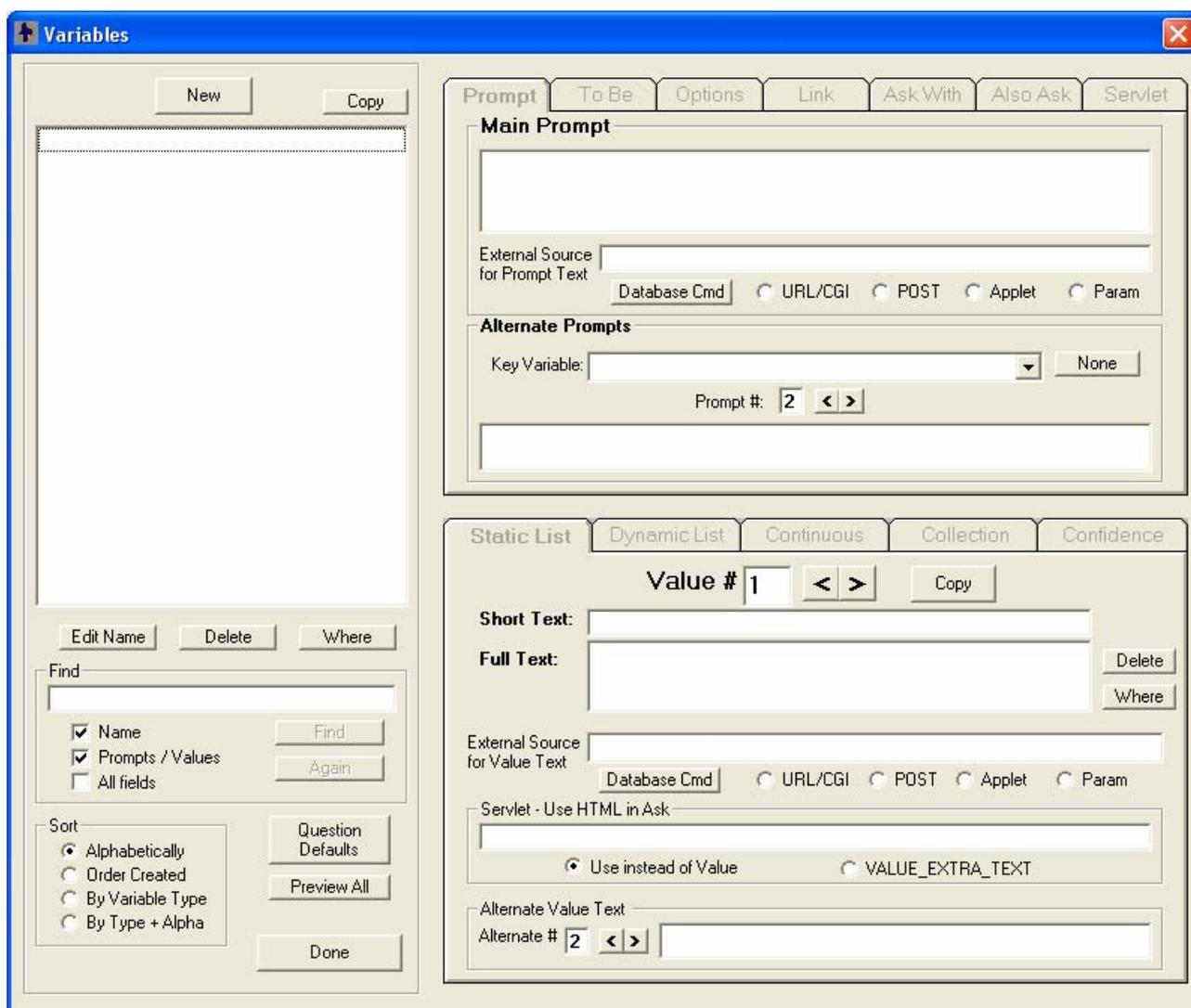
Для создания новой системы выберите пункт New в меню File.



На экране появится стандартное окно сохранения файла – создание новой системы начинается с задания имени файла этой системы. Выберите какую-либо папку для сохранения файлов БЗ и введите имя файла системы, например, BULB. Нажмите кнопку Save (Сохранить).

Шаг 2

Далее на экране появится окно для добавления переменных CORVID:



Все экспертные системы в CORVID используют подобные переменные для определения логики своей работы. Существует несколько типов переменных, в зависимости от типа данных, которые они представляют. Для данной простейшей экспертной системы потребуются только два типа:

1. Static List Variable – переменные со статическим списком значений – переменные, принимающие значения из заранее определенного списка, например, список множественного выбора.
2. Confidence Variable – доверительные переменные – переменные, используемые для представления рекомендаций, которые будут

выдавать система. С доверительными переменными ассоциируется доверительное значение (степень уверенности), которое показывает, основываясь на введенных пользователем данных, насколько достоверна та или иная рекомендация.

Для построения ЭС для выбранного примера, нам понадобятся 3 переменных. Нажмите кнопку New. Отобразится окно для добавления переменных с помощью которого можно добавлять новые переменные определенного типа.

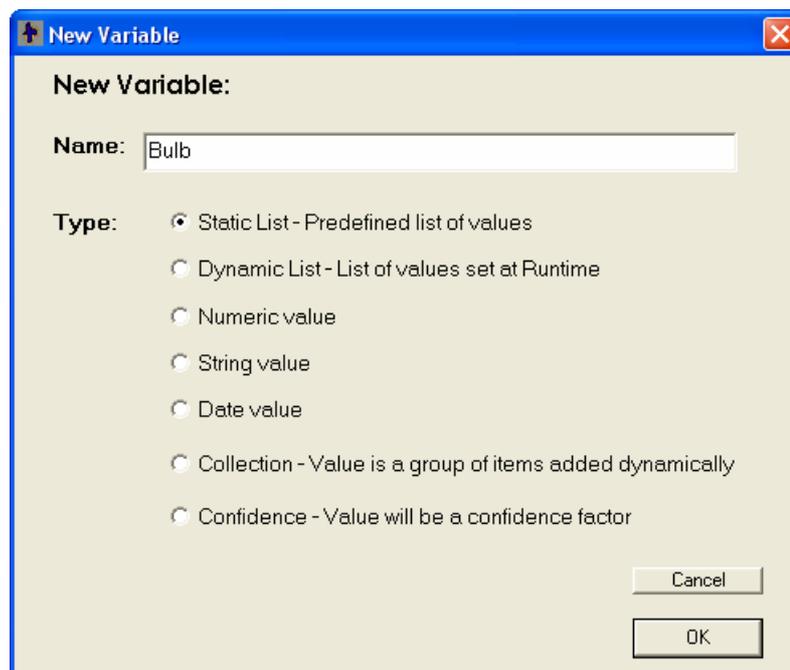
Первое правило новой системы звучит следующим образом:

IF
Свет в Вашем доме внезапно перестал работать
THEN
замените лампочку

Для добавления этого правила в БЗ понадобятся переменные со статическим списком, которые описывают состояние освещения с помощью двух значений:

1. внезапно погас
2. продолжает работать

Переменная должна в данном случае называться «Свет»(Bulb). Имя вводится в поле с меткой Name. Имена переменных могут состоять только из следующих символов: буквы, цифры и символы _#\$%. Все остальные символы из имен переменных удаляются. Пробелы в именах переменных автоматически конвертируются в подчеркивания. Так как тип переменной – переменная со статическим списком, то для ввода значений пользователем предлагается кнопка-переключатель – radio button.



Новая переменная добавляется в список переменных. Комментарий - Prompt - для данной переменной изначально устанавливается такой же, как и имя переменной. Следует изменить комментарий так, чтобы он отражал семантику соответствующей переменной. В данном случае нам необходимо сформировать утверждения «Свет в Вашем доме внезапно погас» и «Свет в Вашем доме продолжает работать». Эти предложения формируются из комментария к переменной и одного из ее значений. Таким образом, комментарий должен быть: «Свет в Вашем доме». Так как комментарий может содержать любые символы, то просто введите данную фразу в поле Main Prompt.

Теперь необходимо добавить возможные значения, входящие в статический список для данной переменной. Для этого введите первое из двух значений «внезапно перестал работать» в текстовое поле Full Text.

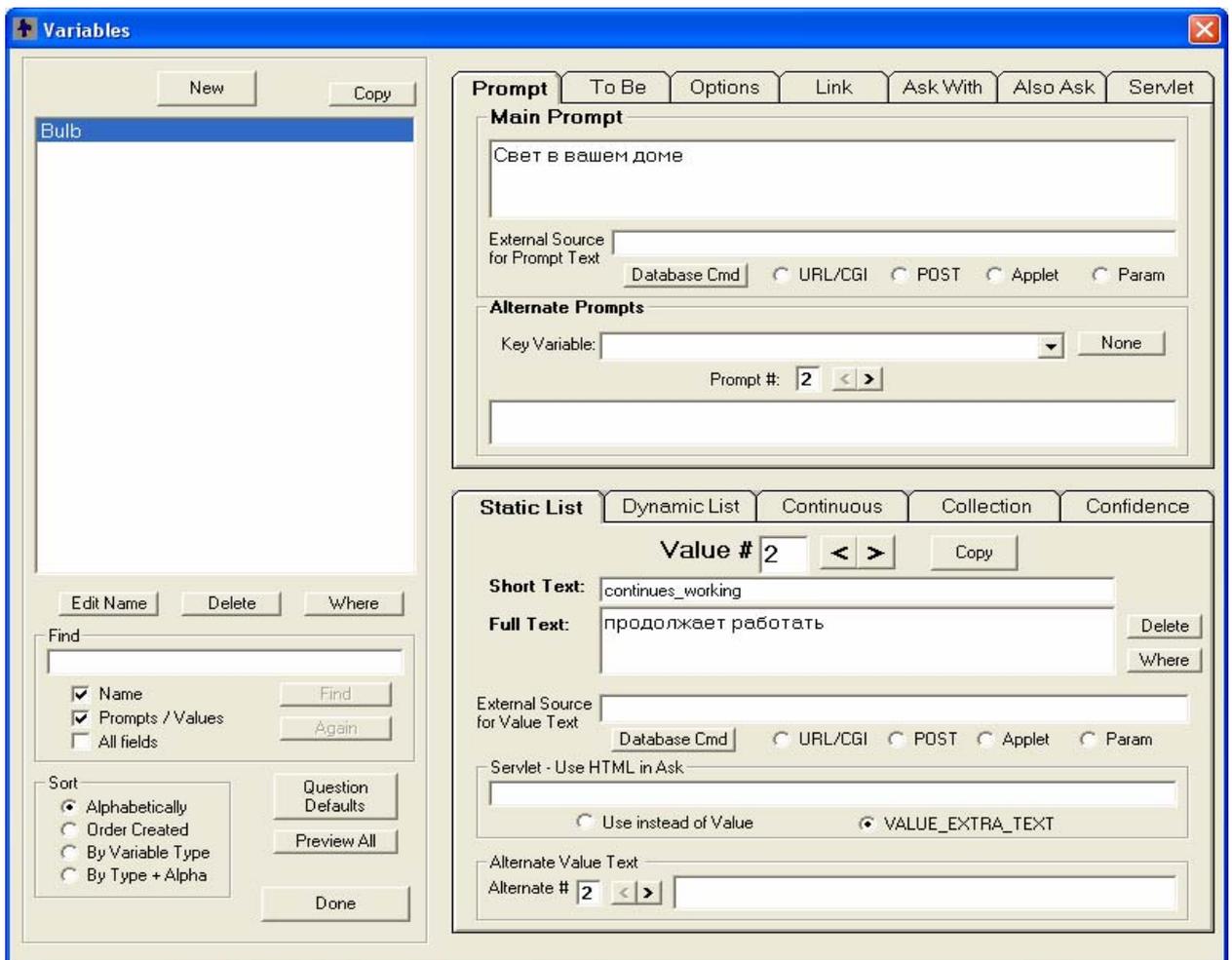
Значения переменной имеют два вида – полный текст (Full text) и сокращенный текст (Short text). Сокращенный текст используется для построения логических блоков и команд, в которых удобнее использовать короткие строки, обычно не более 30 символов. Хотя ограничения на него нет.

Если полный текст значения переменной состоит из небольшого количества символов, то он может быть просто скопирован нажатие клавиши <Enter>. Но часто в качестве сокращенного текста используется подходящий фрагмент полного текста. Для копирования такого фрагмента следует его выделить, в нашем случае это может быть «перестал работать», и нажать кнопку Copy. При этом пробелы автоматически конвертируются в подчеркивания, а неразрешенные символы удалятся из сокращенного текста. (Прим. Поле Short text как и имя переменной заполняются латинскими буквами).

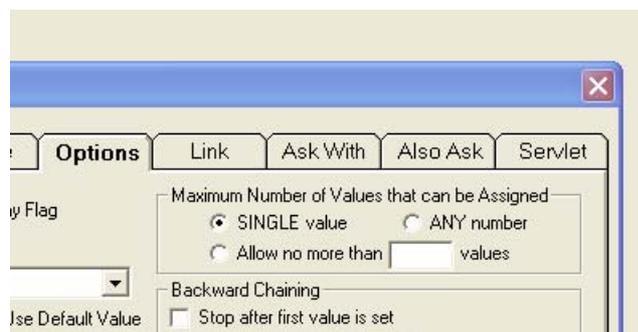
Текст «перестал_работать» появится в поле Short text. Разумеется, можно просто ввести этот сокращенный текст или любую другую часть полного текста с клавиатуры.

Теперь первое значение переменной введено. Для добавления этого значения нажмите кнопку >. Поля Short Text и Full Text очистятся для ввода следующего значения. Кнопки < и > позволяют перемещаться по списку от одного значения к другому. Ограничений на количество значения в списке нет.

Добавьте следующее значение переменной в список. В качестве полного текста введите «продолжает работать» и скопируйте эту фразу в сокращенный текст, нажав клавишу Enter.



Теперь необходимо указать, что пользователь может выбрать только одно значение из списка. Существует несколько способов сделать это в CORVID. В данном случае мы ограничим количество значений, которые могут быть присвоены данной переменной. Для этого выберите закладку Options и выделите кнопку-переключатель «Allow only a single value to be set» - «Позволить устанавливать только одно значение». Выделение этой опции скажет CORVID автоматически использовать кнопки-переключатели – radio buttons – для ввода значений, чтобы исключить выбора нескольких вариантов пользователем.



Первая переменная полностью определена и может быть использована в логических блоках и командах.

Для продолжения построения БЗ нам необходимы еще две переменные, так как правила, которые должны содержаться в БЗ следующие:

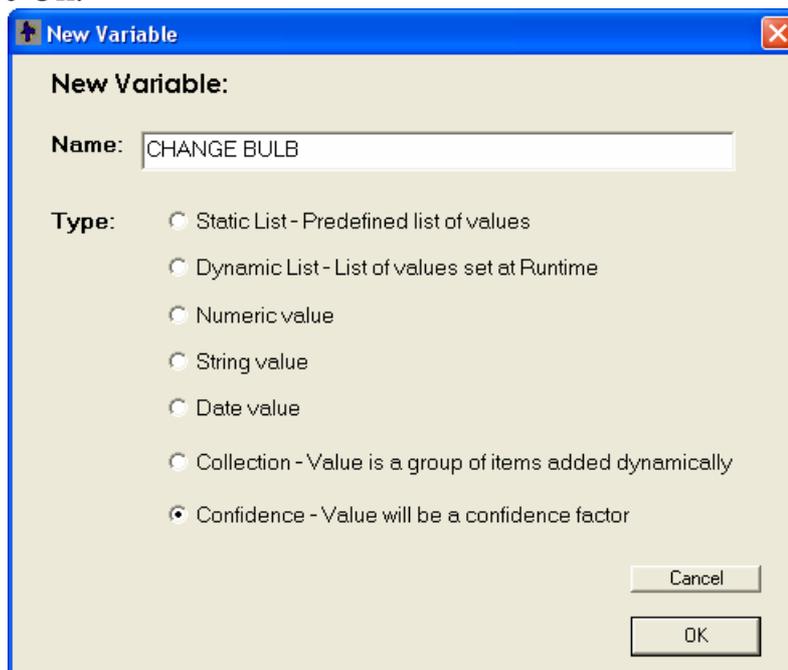
```
IF
    Свет в Вашем доме внезапно перестал работать
THEN
    замените лампочку

IF
    Свет в Вашем доме продолжает работать
THEN
    Ничего не делать
```

IF-части правил уже определены с помощью только что введенной переменной. Теперь необходимо ввести переменные для THEN-частей правил или рекомендаций. В CORVID существует несколько способов определения этих частей правил. Но обычно для целей или действий, которые рекомендует система используются доверительные переменные. Таким образом, для завершения построения БЗ следует добавить 2 доверительных переменных.

Если окно переменных скрыто, то его можно открыть с помощью пункта Variables меню Windows, или с помощью кнопки панели инструментов .

Для добавления переменной нажмите кнопку New. Новая переменная будет использоваться для указания, что необходимо заменить лампочку. Поэтому назовем ее «Заменить_лампочку». И необходимо указать тип переменной – Confidence, выбрав соответствующую кнопку-переключатель. Затем нажмите Ok.



Для новой переменной автоматически будет установлен комментарий, повторяющий ее имя. Замените его на более подходящий для сообщения, которое будет выдано пользователю, например, «Замените лампочку».

Опции для доверительной переменной устанавливаются на закладке Confidence – значения, которые она может принимать и алгоритмы комбинирования этих значений, для вывода окончательного значения.

Для данного примера можно оставить значения по-умолчанию, а именно, входные значения не ограничены, и окончательное значение вычисляется как сумма всех введенных чисел. Таким образом, на закладке Confidence следует оставить все без изменений.

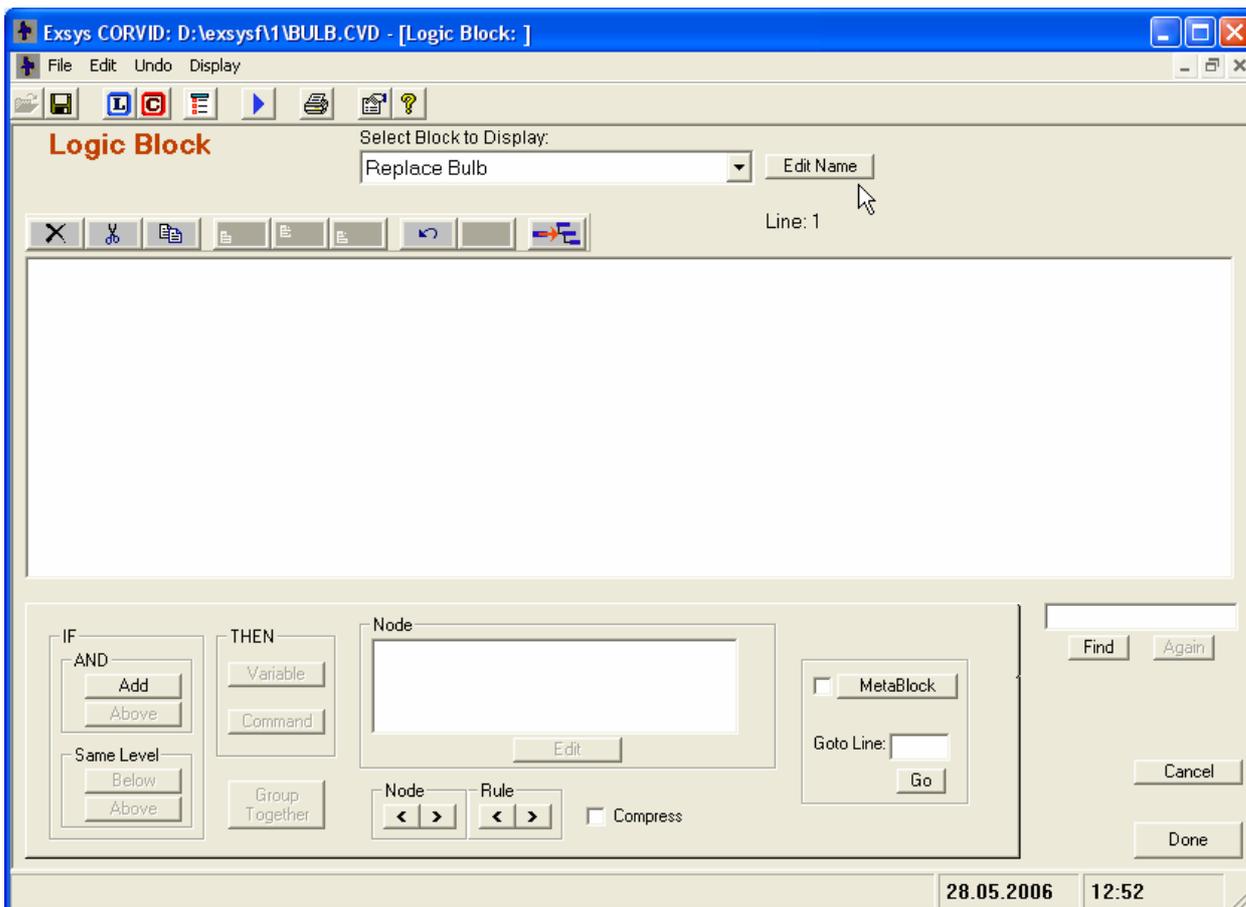
Аналогично добавьте переменную «Ничего_не_делать» – нажмите кнопку New, введите имя переменной, выберите тип переменной Confidence и нажмите кнопку Ok.

Измените комментарий для новой переменной на «Ничего не делать». Теперь нажмите кнопку Done, чтобы запомнить изменения и закрыть окно Variables.

Шаг 3

Теперь, когда все необходимые переменные определены, следует перейти к построению логических блоков. С помощью логических блоков описываются знания в системе. Логический блок может содержать одно или несколько логических деревьев и/или правила. Вся система может быть построена с помощью нескольких логических блоков. Отдельные логические блоки позволяют сгруппировать деревья и правила, относящиеся к какому-либо аспекту решаемой проблемы.

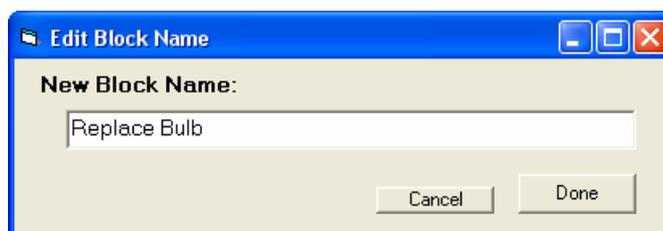
Для добавления логического блока, выберите пункт Logic Block меню Windows или нажмите кнопку панели инструментов . На экране появится окно Logic Block.



В этом окне выполняется построение и редактирование логических блоков. Просматривать содержимое логического блока можно также в окне правил – Rule View, которое отображает БЗ в виде правил.

Когда открывается новое окно Logic Block, оно получает имя «Logic Block #», где # – это номер логического блока. Изменять это имя не обязательно, но если в системе используется более одного логического блока, то лучше переименовать логический блок так, чтобы его имя отражало решаемую подзадачу.

В нашем примере изменим название логического блока на «Замена лампочки». Для этого нажмите кнопку Edit Name. В появившемся окне введите «Замена лампочки» вместо «Logic Block 1».

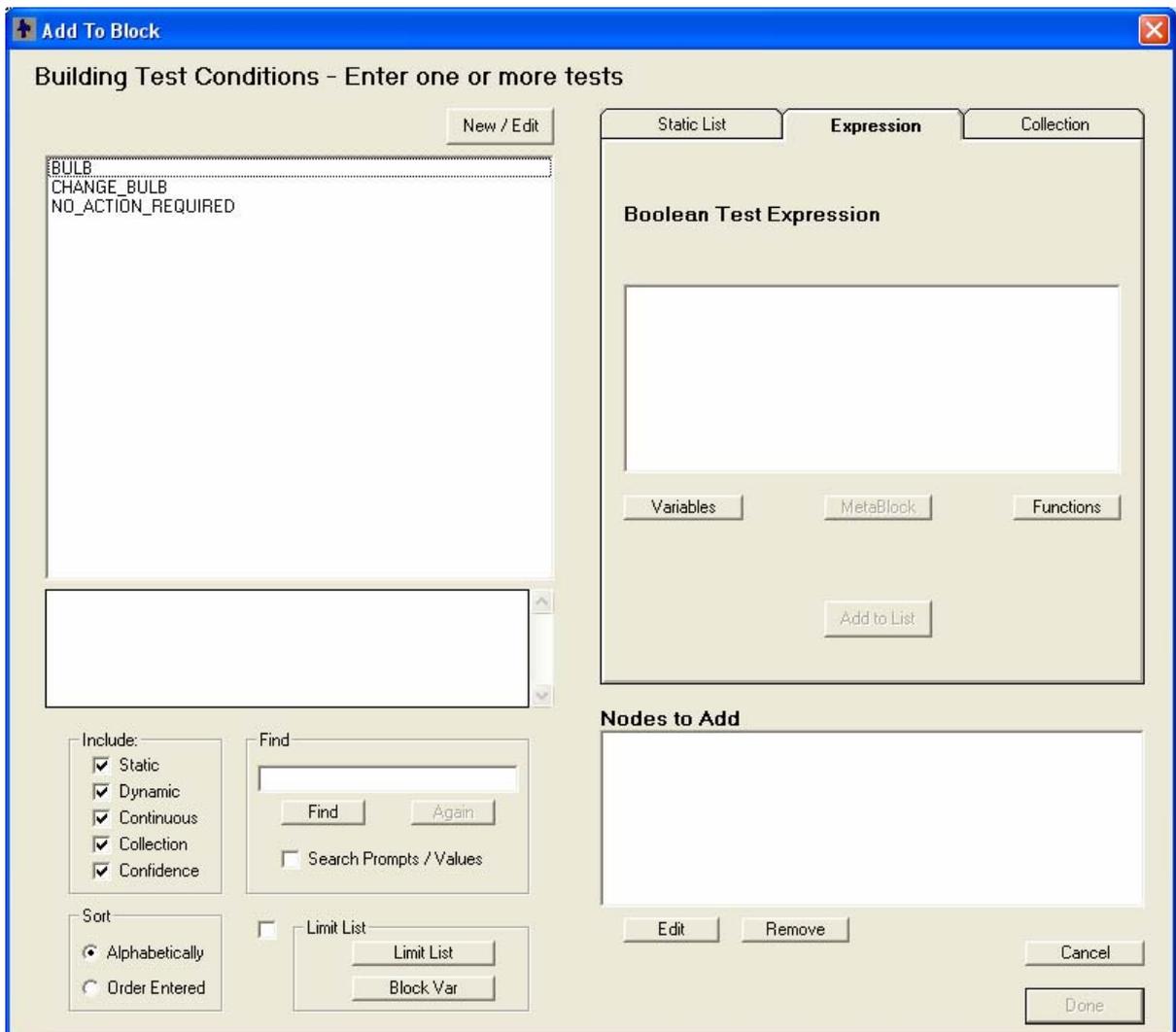


Вспомним, что правила, которые необходимо добавить в БЗ выглядят так:

IF
 Свет в Вашем доме внезапно перестал работать
THEN
 замените лампочку

IF
 Свет в Вашем доме продолжает работать
THEN
 Ничего не делать

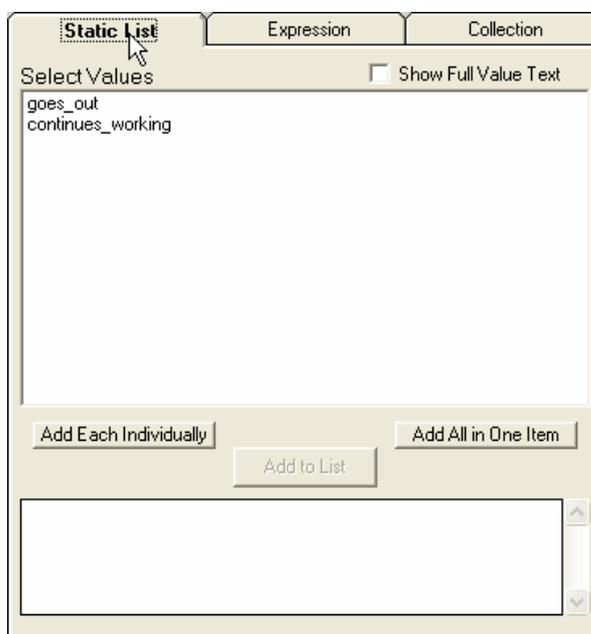
Для начала следует добавить IF-части правил. В данном примере имеются 2 правила, и условия этих правил относятся к состоянию одного и того же объекта (лампочки или света). Это наводит на мысль, что можно использовать логическое дерево для описания данных знаний. Древовидные структуры позволяют отслеживать, что все возможные варианты значений учтены в правилах, их быстрее построить и легче поддерживать, чем отдельные правила (хотя последние и могут использоваться вместо деревьев).



Если логический блок пустой, то активна только кнопка ADD IF. Заметьте, что всего имеется 4 кнопки для добавления IF-части правила. Они позволяют добавлять If-части в различные позиции относительно текущего выделенного узла дерева. Как правильно использовать эти кнопки станет яснее, когда дерево будет уже частично сформировано. Сейчас просто нажмите кнопку IF /AND / Add.

На экране отображается окно для добавления узлов к логическому блоку – Add To Block. С помощью данного окна могут быть добавлены различные типы узлов. Но так как для IF-части правила «Свет в Вашем доме внезапно перестал работать» переменная уже добавлена, то для выделения требуемой переменной просто щелкните мышкой на имени «Свет».

Переменная «BULB» (СВЕТ) выделена. В правой части окна появляется статический список значений для выделенной переменной. Список значений отображаются на закладке Static List.



Так как первое выражение для If-части правила звучит «Свет в Вашем доме внезапно перестал работать», то следует выбрать значение «goes_out» (перестал работать). В качестве значения отображается сокращенный текст переменной. Чтобы увидеть полный текст, взведите галочку Show Full Value Text.

Однако, нам следует также сформировать If-часть для другого правила: «Свет в Вашем доме продолжает работать». Таким образом, следует добавлять отдельные ветви дерева для каждого из двух значений. Наиболее быстрый способ это сделать – нажать кнопку Add Each Individually.

Это можно также сделать, нажимая кнопку Add to List последовательно для каждого значения. Но первый способ проще и быстрее.

В список Nodes to Add внесены два оба требуемых условия для будущих правил. По мере добавления в список Nodes to Add значения удаляются из списка Select Values. Так как была нажата кнопка Add Each

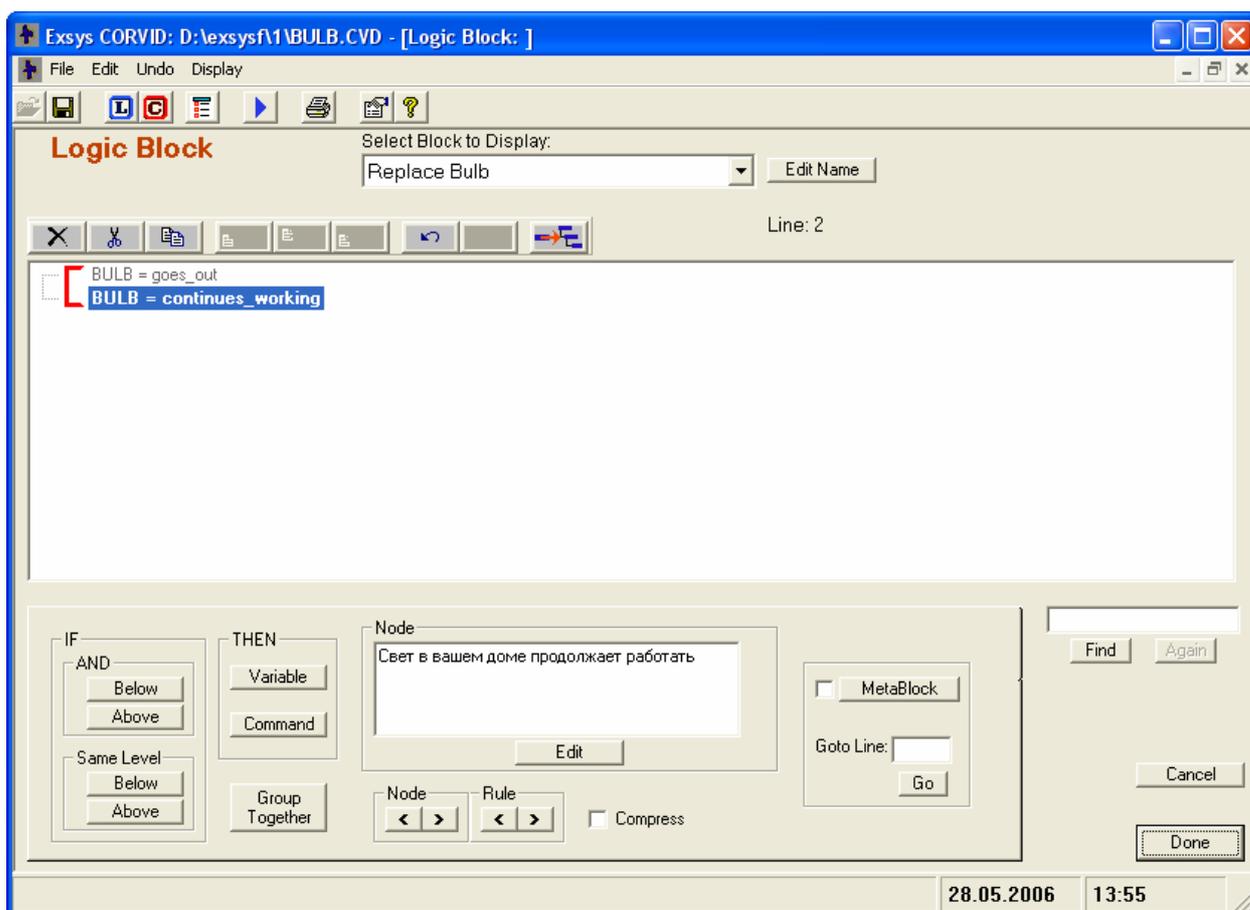
Individually, все значения переместились из верхнего списка в нижний одновременно.

Выполненная только что операция позволяет создать для каждого значения индивидуальный узел в дереве. Однако, иногда требуется объединить несколько значений, схожих по смыслу в один узел. Это можно сделать, выделив несколько требуемых значений с помощью комбинации клавиши shift или ctrl + щелчок мыши и нажав кнопку Add All in One Item.

В нашем примере необходимые узлы уже сформированы, поэтому можно нажать кнопку Done.

Теперь в окне Logic Block добавлены 2 If-узла, которые будут использованы для формирования If-частей будущих правил. Эти узлы объединены скобкой красного цвета. Эта скобка указывает, что узлы добавлены одновременно и взаимосвязаны между собой. Обычно это показывает, что узлы построены по различным значениям одной переменной. Если бы мы добавили больше узлов, то они также были бы связаны скобкой.

Такие скобки особенно полезны, когда необходимо улучшить систему, так как они отображают взаимосвязанные узлы. Однако, эти скобки лишь повышают информативность отображаемой информации и ни как не влияют на логику работы системы.

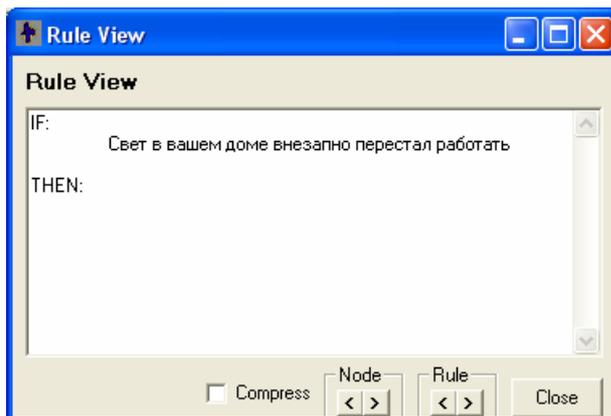


Красный цвет скобки показывает, что с данными If-узлами не ассоциированы никакие Then-узлы. То есть они не формируют пока еще полноценные правила. Когда недостающие then-узлы будут добавлены,

скобки изменяют свой цвет на зеленый. Таким образом, красные скобки помогают отслеживать состояние дерева, если оно достаточно сложное и разветвленное.

Теперь, чтобы добавить узел THEN для первого узла «Свет в Вашем доме внезапно перестал работать» сперва необходимо его выделить, щелкнув мышкой на строке «BULB = goes_out» (Свет перестал работать).

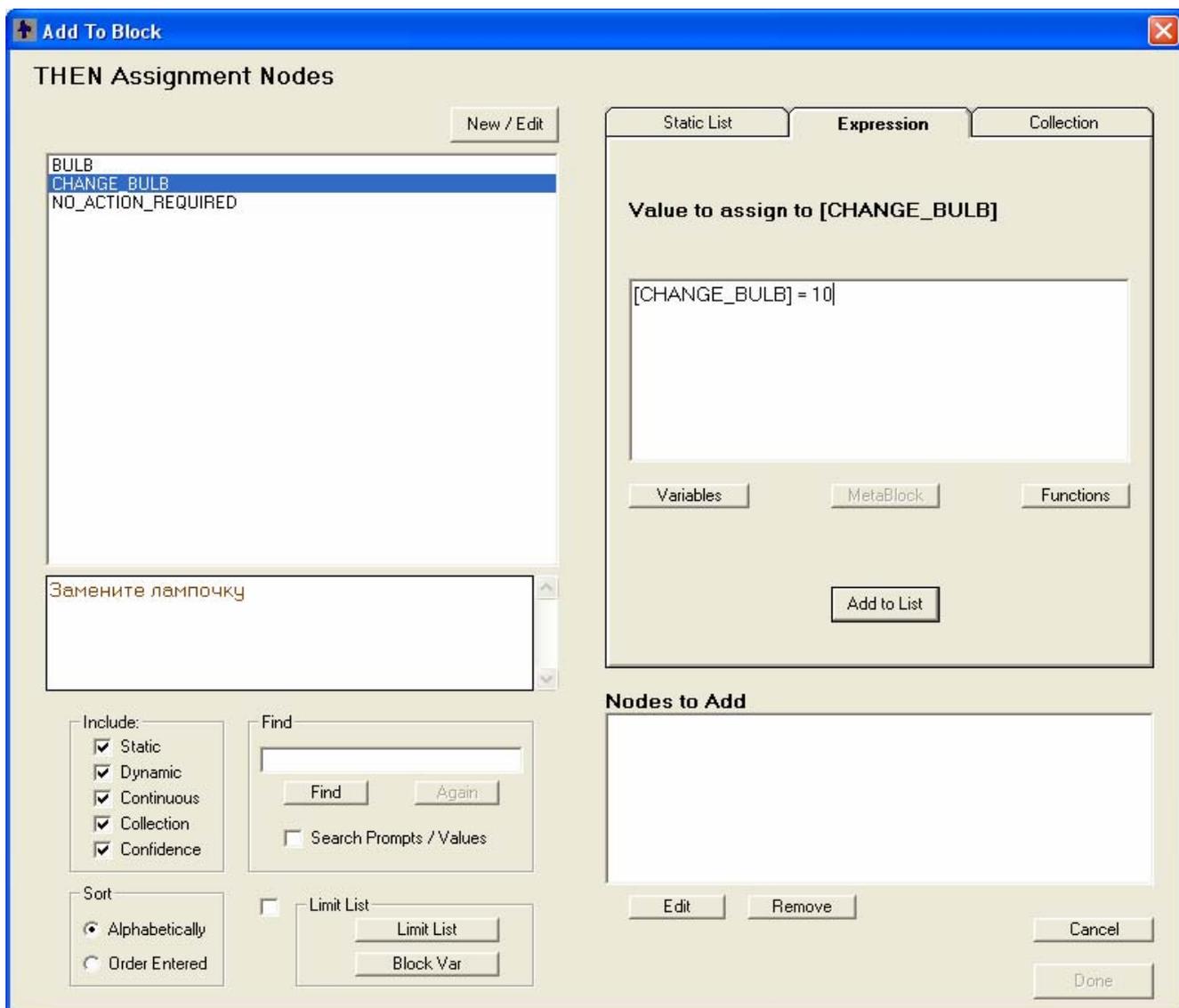
Как только будет выделен узел «BULB = goes_out» (Свет перестал работать), в окне Rule View отобразится правило с полным текстом значений для данного узла.



Недостающий узел THEN может быть построен либо на основе присваивания значения какой-либо переменной, либо с помощью команды. (Команды реже используются в логических блоках). Для добавления Then-узла на основе переменной нажмите кнопку Variable, расположенной под меткой THEN. Снова появится окно добавления узлов в логический блок.

THEN-узлы всегда присваивают какие-либо значения переменным. В данном примере мы хотим добавить узел, соответствующий знанию о том, что следует заменить лампочку. В систему уже добавлена доверительная переменная представляющее это действие – «CHANGE_BULB» (Замените лампочку). По-этому щелкните не имени этой переменной, чтобы выделить ее.

Доверительные переменные имеют много применений в CORVID, но наиболее часто они используются для представления возможных рекомендаций ЭС. Доверительные значения, присеваемые переменным, также могут использоваться различными способами для построения вероятностный систем. Однако в данном случае, мы просто присвоим переменной фиксированное значение (например, 10) и используем это для индикации того, что эта переменная будет являться рекомендацией.



Результирующий отчет будет содержать все доверительные переменные, получившие значения больше 0. Таким образом, мы можем присвоить любое положительное значение переменной. Выберем для переменной «CHANGE_BULB» (Замените лампочку). значение 10. Нажмите кнопку Add to List, чтобы переместить эту переменную в список узлов для добавления Nodes to Add.

Обратите внимание: в большинстве случаев переменные в CORVID отображаются в квадратных скобках []. Переменные со статическим списком значений отображаются без скобок, чтобы их легче было читать.

Составные THEN-узлы могут добавляться таким же образом. Однако, они должны быть ассоциированы с одним и тем же If-узлом. Поскольку в данном случае нам требуется одиночных THEN-узел, то нажмите кнопку Done для добавление узла к логическому блоку.

Теперь узел [CHANGE_BULB]=10 добавлен к дереву в логическом блоке. Этот узел отмечается маленькой стрелкой. Желтый цвет указывает на то, что этот узел построен с помощью доверительной переменной. Отступ перед новым узлом относительно узла «BULB=goes_out» указывает на то,

что эти узлы связаны. Если добавляется составной THEN-узел, то все его элементы отображаются подобным образом.

Соответствующее выделенному узлу дерева правило IF/THEN также отображается в окне Rule View.

Теперь добавим второе правило.

IF	Свет в Вашем доме продолжает работать
THEN	Ничего не делать

Выделите IF-узел «BULB=continues_working» (Свет продолжает работать). Также как и в прошлый раз следует присвоить положительное значение соответствующей доверительной переменной. По-этому нажмите кнопку Variable по меткой THEN.

В появившемся окне Add to Block выполните следующие действия:

1. Выделите «NO_ACTION_REQUIRED» (Ничего не делать).
2. Введите значение 10, которое будет присвоено выделенной переменной.
3. Нажмите кнопку Add to List.

После этих действий в списке Nodes to Add появиться имя выделенной переменной [Ничего_не_делать] и присвоенное ей значение. Нажмите кнопку Done для добавления нового узла в логический блок.

Второй THEN-узел добавлен в дерево логического блока. Соответствующее ему правило отображается в окне Rule View.

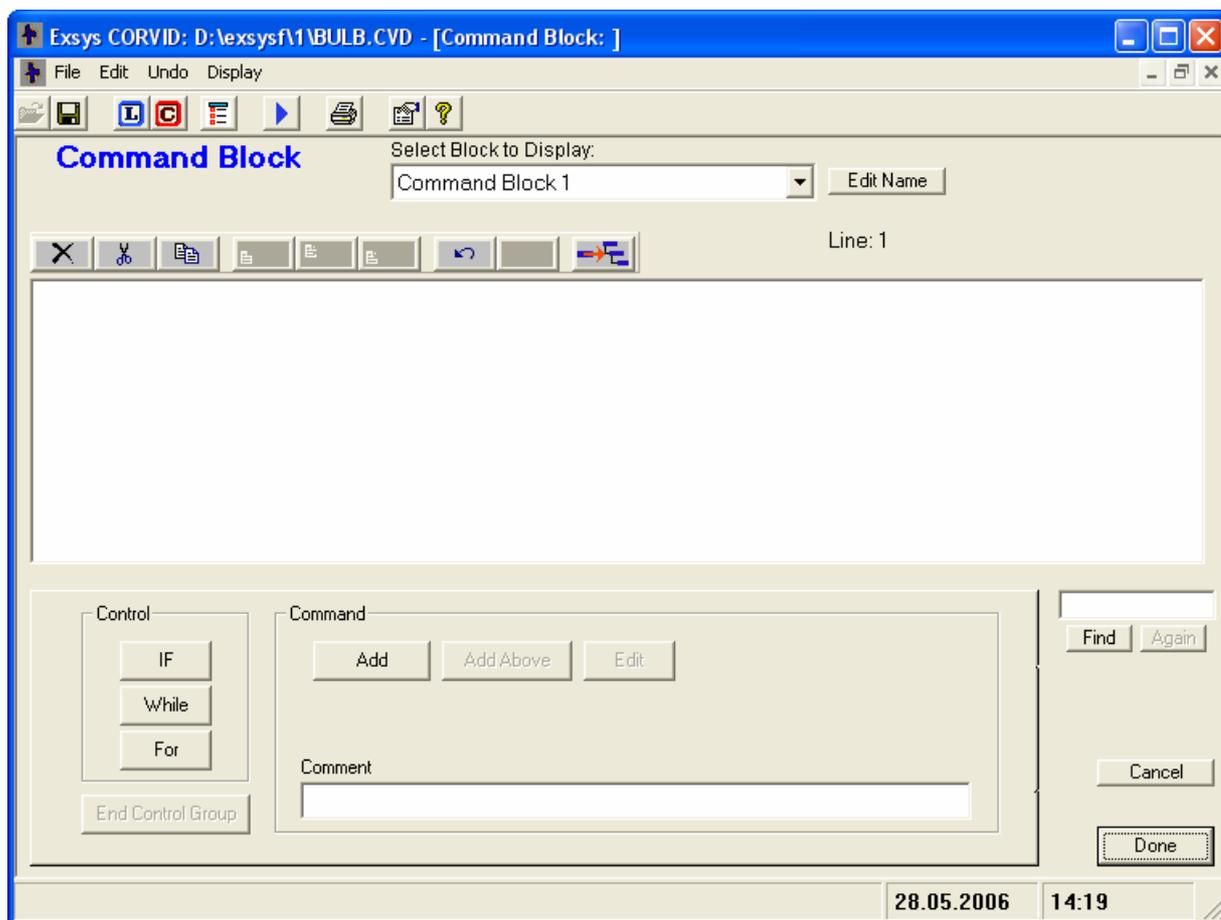
Для данной простейшей системы добавлены все необходимые узла для построения двух требуемых правил. В последующих лабораторных работах возможности этой системы будут расширены.

Поскольку логический блок еще не завершен, нажмите кнопку Done для закрытия окна.

Шаг 3 Добавление Командных блоков

БЗ, разрабатываемые в CORVID, должны содержать командные блоки. Если логические блоки говорят системе о том как принять решение, то командные блоки описывают что нужно сделать. Другими словами, командный блок определяет цель, а логический блок путается ее достичь.

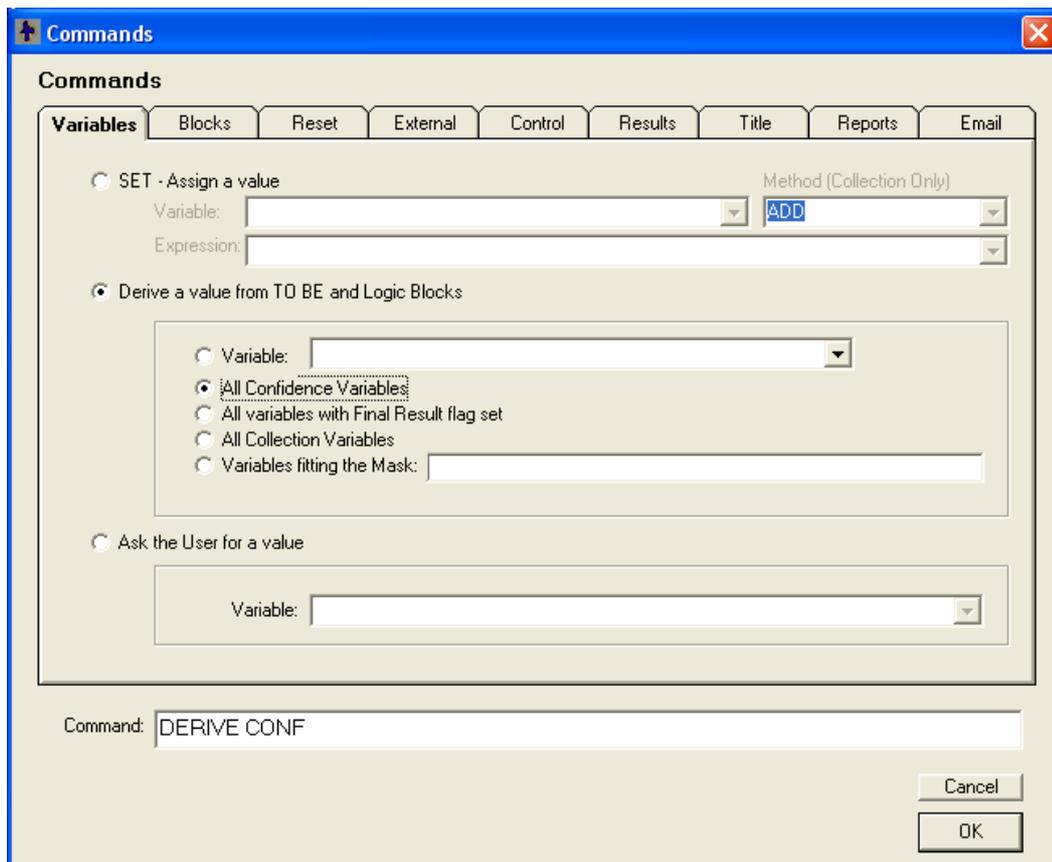
Для добавления командного блока выберите пункт Command Block в меню Windows или нажмите кнопку в панели инструментов. На экране появиться окно для добавления командных блоков.



По-умолчанию командному блоку назначается имя Command Block 1. Большинство систем используют только один командный блок, по этому имя можно не изменять. Для добавления нового блока нажмите кнопку Add. На экране появится окно формирования командного блока Commands.

Это окно используется для построения любых команд в CORVID. Вам не нужно запоминать синтаксис этих команд, просто выберите соответствующие опции. В логическом блоке были использованы доверительные переменные для представления возможных рекомендаций системы. Следовательно, необходимо добавить в систему команды для указания системе использовать эти логические блоки для вывода значений доверительных переменных.

Для этого необходимо перейти на закладку Variables и нажать кнопку переключатель Derive a value from TO BE and Logic Blocks. В соответствующей секции можно указать конкретные переменные или группу либо по их типу либо по текстовому шаблону. В данном случае нам необходимо вывести значения всех переменных, поэтому выделите переключатель All Confidence Variables.

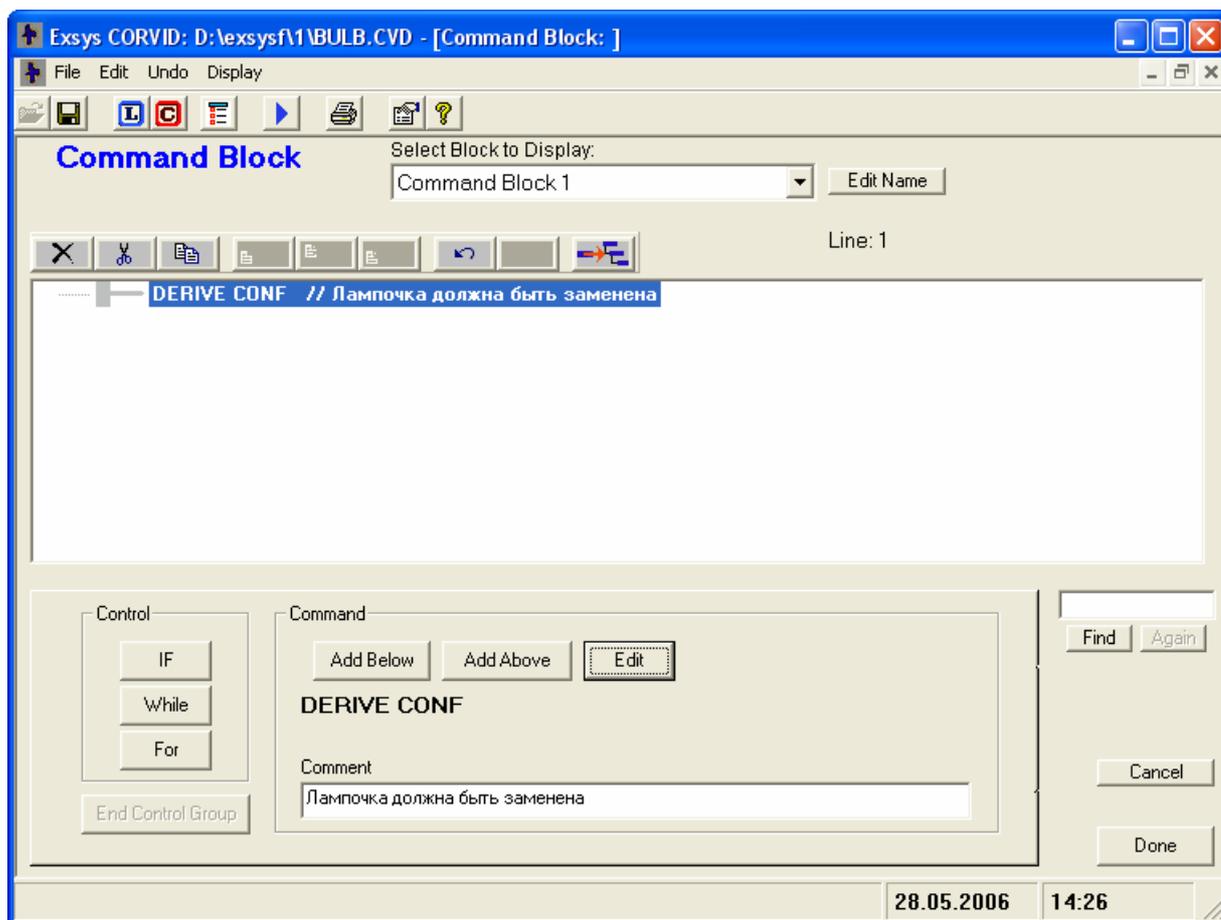


В данном примере мы сформировали команду для запуска системы. В различных системах могут использоваться разные типы команд для запуска. Так как большинство систем используют доверительные переменные для описания возможных целей, то приведенная команда DERIVE CONF используется очень часто. Для добавления этой команды нажмите кнопку ОК.

В командный блок добавлена новая команда. К каждой команде добавляется комментарий. Эти комментарии предназначены для документирования команд и облегчают поддержку системы. Для добавления комментария просто введите его в поле редактирования Comment. В данном примере введите «Лампочка должна быть заменена», чтобы обозначить, что команда устанавливает переменную для этого решения.

После ввода текста, комментарий автоматически добавляется к команде. Комментарии не являются обязательными и могут не добавляться к командам.

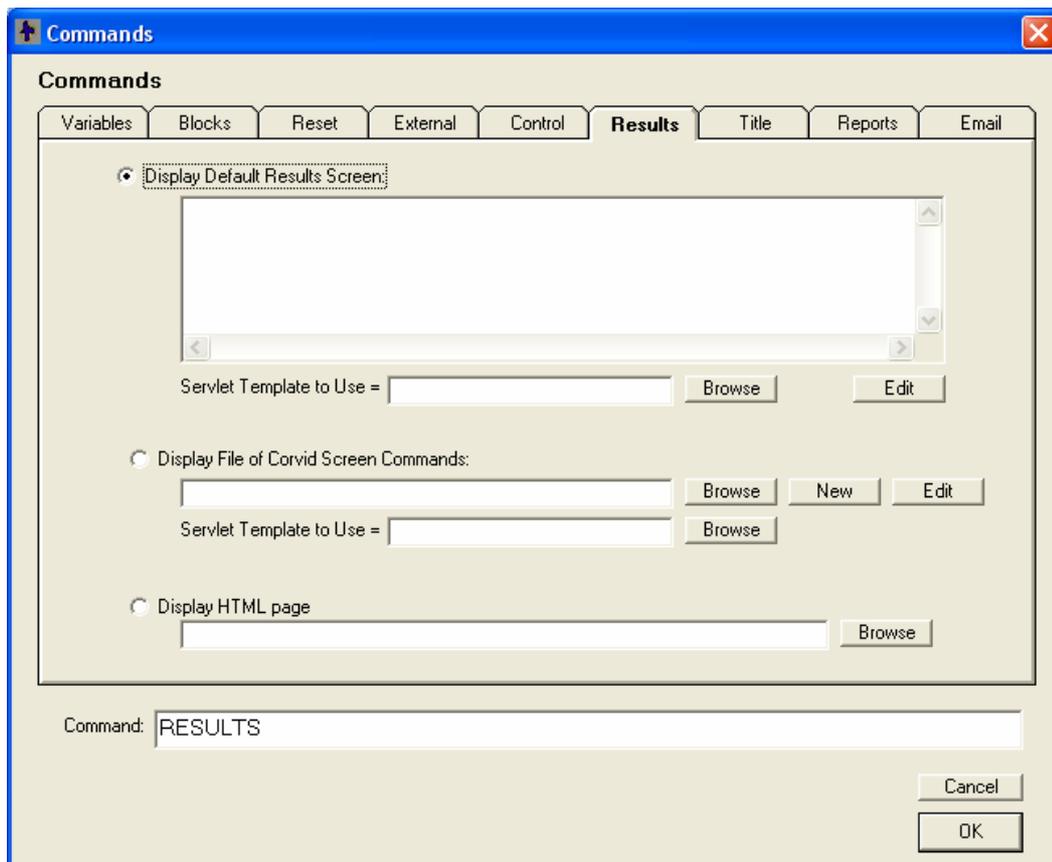
Команда DERIVE CONF позволит осуществить вывод значений всех доверительных переменных с помощью логического блока. Теперь необходимо отобразить результаты работы системы. Для этого используется команда RESULTS, которая также добавляется с помощью окна формирования команд.



Если команда DERIVE CONF не выделена, то выделите ее. Затем нажмите кнопку Add Below для добавления новой команды. (Кнопки Add Below и Add Above добавляют новые команды после или перед текущей выделенной командой). Снова появилось окно Commands в котором необходимо перейти на закладку Results.

Существует несколько способов отображать результаты. Наиболее общий способ – это использовать стандартное окно результатов. В законченных системах следует вводить команды отображения результатов, которые определяют, какую и как информацию следует отображать и в каком формате. Чтобы использовать стандартную форму отображения выберите вариант Display Default Results Screen.

В данной системе нет команд для отображения результатов, поэтому на экран просто будут выведены все доверительные переменные, которым было присвоено значение в процессе выполнения работы ЭС. В следующих работах форма отображения результатов будет изменена. После выбора Display Default Results Screen нажмите кнопку OK.



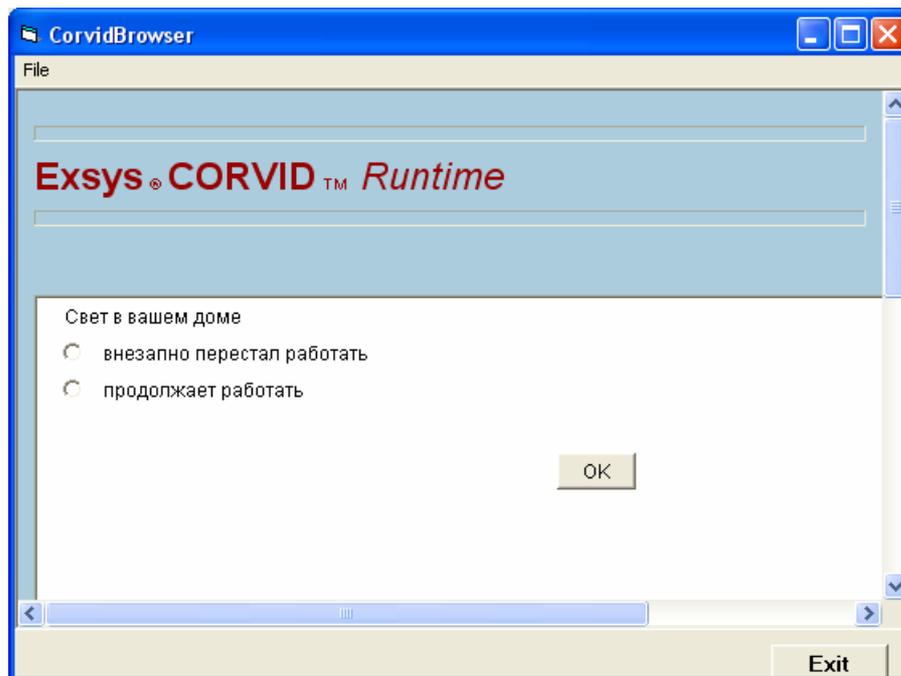
Еще одна команда добавлена в командный блок. Так как смысл команды RESULTS очевиден, то нет необходимости добавлять комментарий.

Теперь все команды, которые необходимы для работы данной системы сформированы, поэтому нажмите кнопку Done.

Шаг 4 Запуск системы

Теперь, когда система построена, мы можем запустить ее. Для запуска системы нажмите кнопку панели инструментов.

Команда Run открывает новое окно. В действительности это Internet Explorer, открытый в меньшем размере и без меню, кнопок и пр. При запуске системы CORVID автоматически формирует специальные .CVR файлы, которые используются для отображения интерфейса пользователя с помощью Java-апплетов. Также формируется HTML-страница, которая содержит вызовы этих Java-апплетов. Именно эта страница отображается в браузере во время работы с пользователем.



Система начинает работу с выполнения первой команды в командном блоке. В данном случае это команда вывести значения всех доверительных переменных. Для этого системе необходимо активировать правила логического блока. Для проверки истинности правил системе необходимо узнать о состоянии освещения. Если есть какие-либо другие правила, позволяющие судить об этом, то система использует их. Однако, в данной системе единственный способ получить необходимую информацию – это спросить у пользователя. Поэтому первое окно содержит вопрос о состоянии освещения.

Выберите «внезапно перестал работать» и нажмите кнопку ОК. Этой информации достаточно, чтобы для присвоения значения доверительной переменной.

Следующая команда в командном блоке – это RESULTS. Так как в данную команду не включены директивы по представлению и форматированию результатов, система отображает значения всех переменных, которым было присвоено какое-либо значение. Результирующее окно предлагает заменить лампочку с доверительным значением 10.

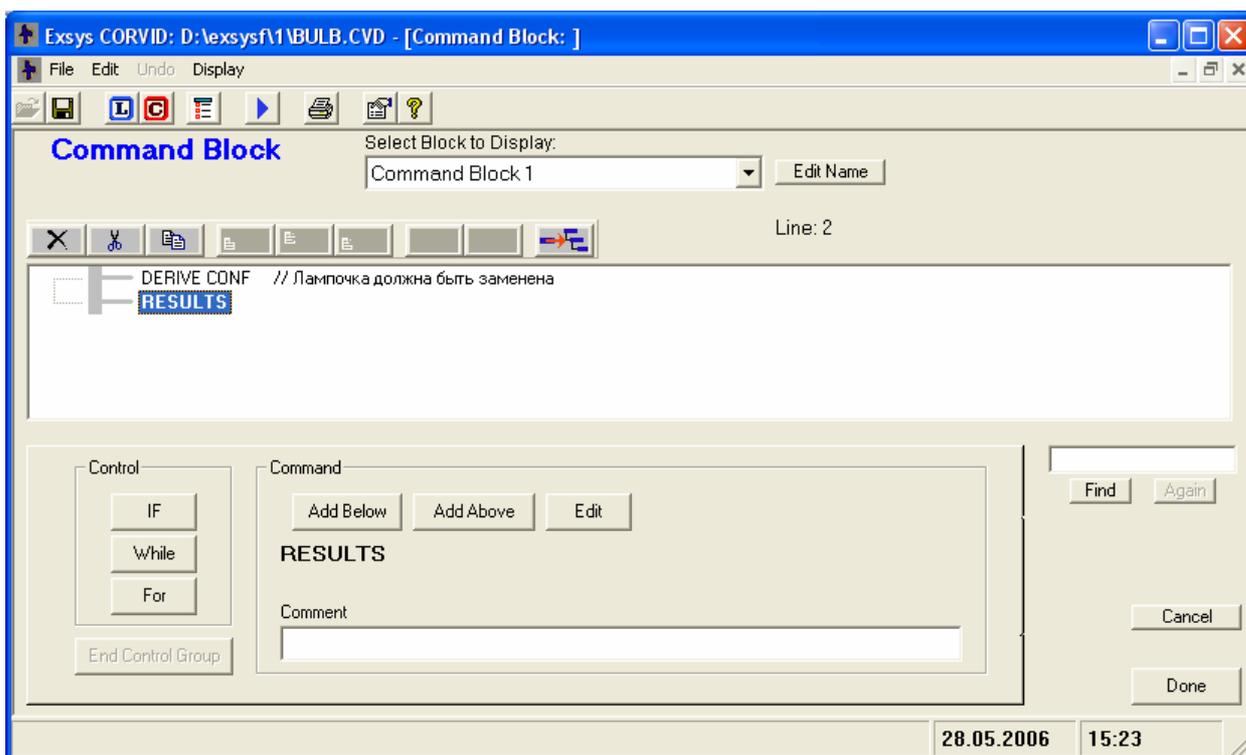
Лабораторная работа 2: Улучшение интерфейса пользователя

Цель работы: изучение возможностей форматирования интерфейса пользователя в системе Exsys CORVID.

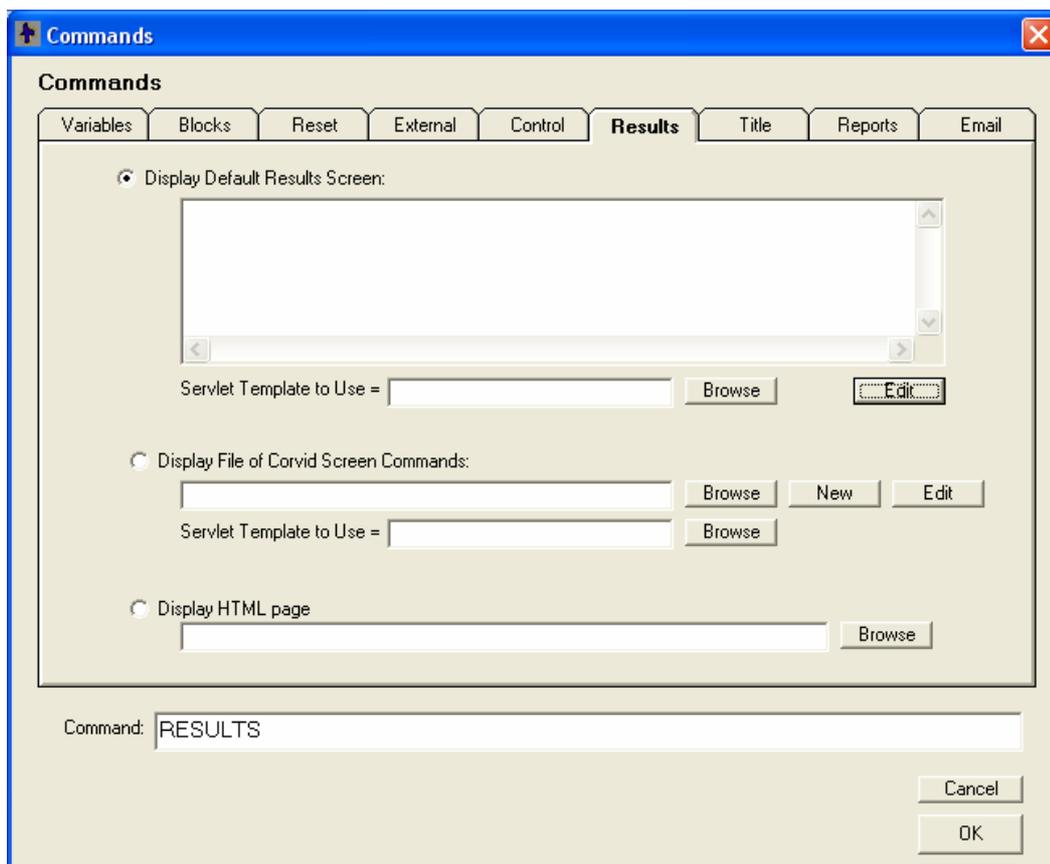
Для изменения способа представления результатов работы системы необходимо добавить специальную команду отображения. Это можно сделать двумя способами: выбрать пункт Set Results Default в меню Windows, или изменить созданную Вами команду RESULTS.

Чтобы создать новую команду следует:

1. открыть окно редактирования командного блока, нажав кнопку
2. В открывшемся окне Command Block выбрать имя ранее определенного блока Command Block 1 в разворачивающемся списке.
3. Выделить команду RESULTS и нажать кнопку Edit.



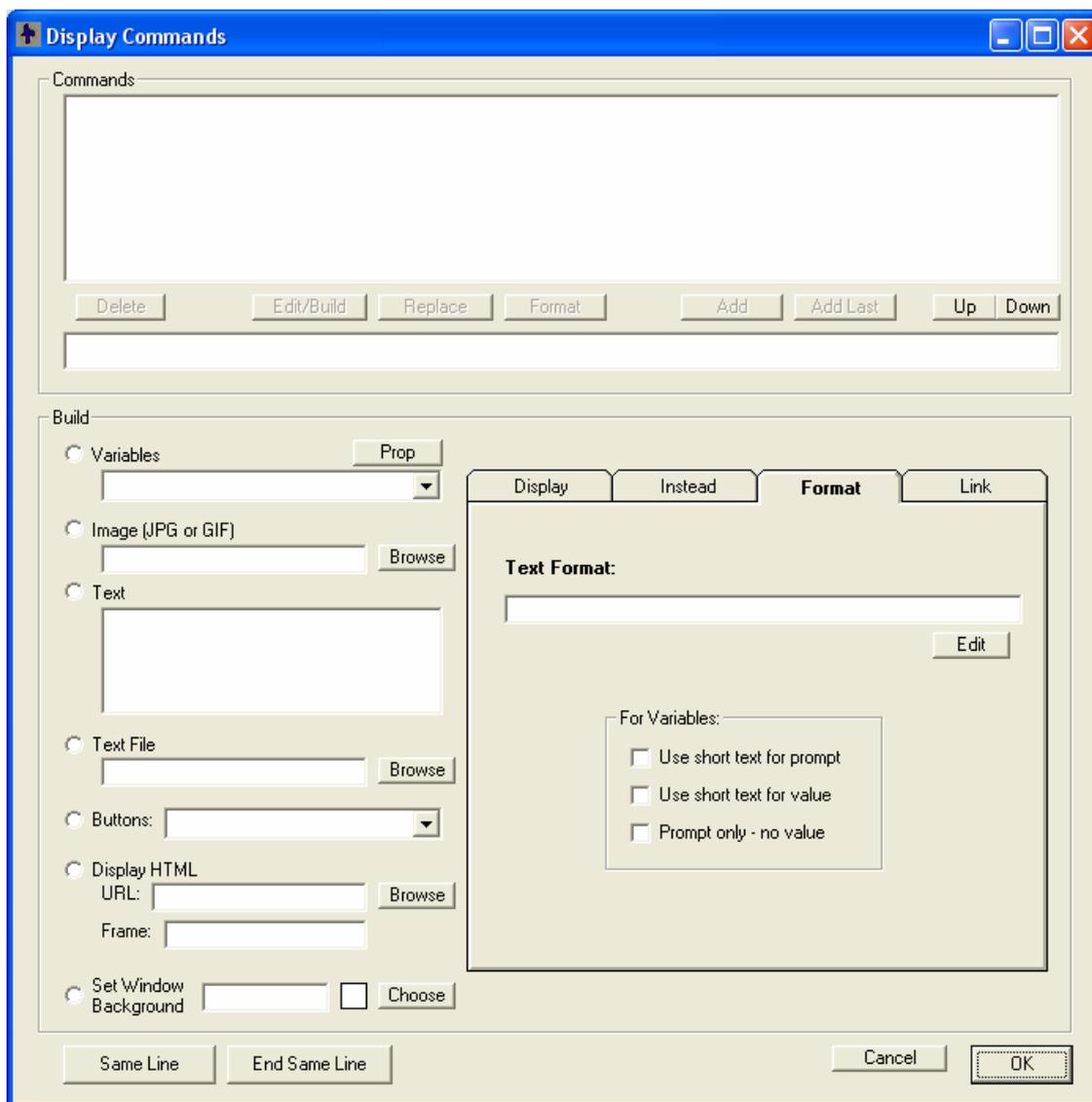
Редактирование команды RESULTS осуществляется в окне Commands. Отметьте, что в списке на закладке Results не содержится никаких команд, ассоциированных с RESULTS. В такой ситуации в результирующем окне отображаются все переменные, получившие в ходе вывода какие-либо значения, что и происходило в работе лабораторной 1. Для добавления специфических команд отображения результатов нажмите кнопку Edit на закладке Results.



Команды отображения создаются в окне Display Commands. В этом окне можно создавать формы отображения, включающие текст, графику, файлы и значения переменных в CORVID.

В верхнем списке показываются добавленные команды (в данном случае он пуст). Команды создаются с помощью элементов управления, расположенных в нижней части окна и отображаются в поле редактирования в верхней части окна. Добавление команды в список осуществляется с помощью кнопки Add. Команды в списке могут быть удалены или изменены с помощью кнопок Delete и Edit соответственно.

Окно результатов должно содержать только те доверительные переменные, которым были присвоены значения. Существуют различные способы добавления переменных CORVID в окно результатов: по одной или по типу. В данном случае нам необходимо добавить все доверительные переменные. Для этого в разворачивающемся списке Variables выберите CONFIDENCE.



Команда отображения CONFIDENCE теперь выделена и отображается в поле редактирования. Пусть нам необходимо отображать только текст (prompt) доверительных переменных без их значений. Для этого необходимо задать форматирование.

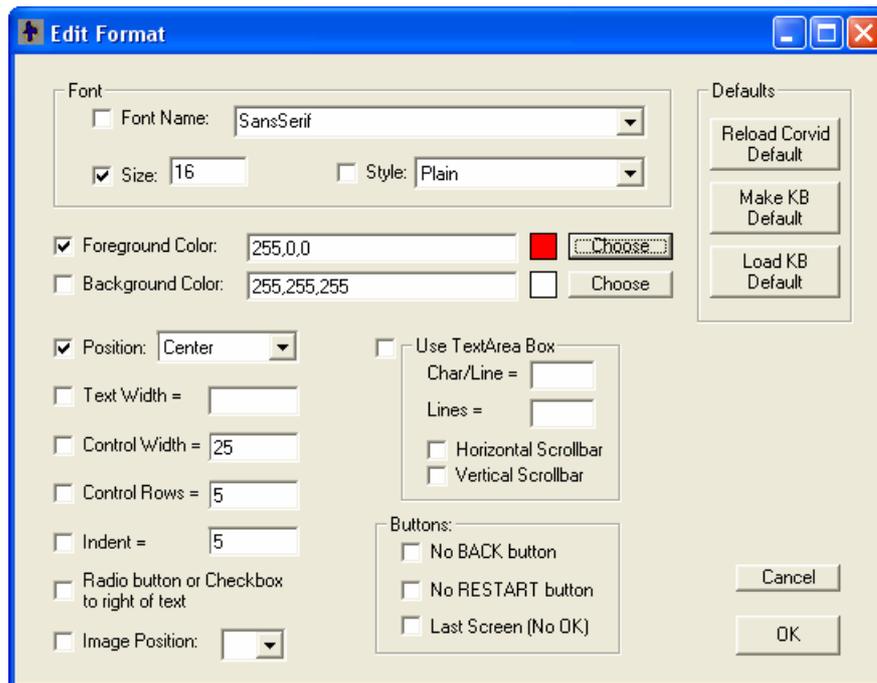
Форматирование определяется на закладке Format, где предлагается выбрать различные способы отображения переменных. Для выбора требуемого в нашем случае варианта взведите галочку рядом с меткой Prompt only – no value.

Далее можно определить шрифт, выравнивание и пр. для отображения на экране. Для этого нажмите кнопку Edit на закладке Format, вследствие чего на экране появится окно форматирования отображения текста.

Предположим, требуется выводить текст размером 16 pt, центрированным и красным цветом. Для этого следует:

1. В поле Size ввести значение 16,
2. В разворачивающемся списке Position выбрать Center,
3. Нажать кнопку Choose рядом с меткой Foreground Color и выбрать красный цвет в стандартном окне Windows для редактирования цвета.

4. Нажать ОК для применения формата.



Теперь в поле ввода содержится команда отображения с введенными параметрами форматирования. Нажмите кнопку Add для добавления этой команды в список команд.

И кнопку ОК, чтобы завершить ввод команды, закрыть окно Display Commands и вернуться в окно Commands.

С командой RESULTS теперь ассоциирована команда отображения, которая отображается в списке на закладке Results. Для продолжения работы нажмите кнопку ОК.

В окне Command Block нажмите кнопку Done.

Для тестирования выполненных изменений запустите систему с помощью кнопки Run.

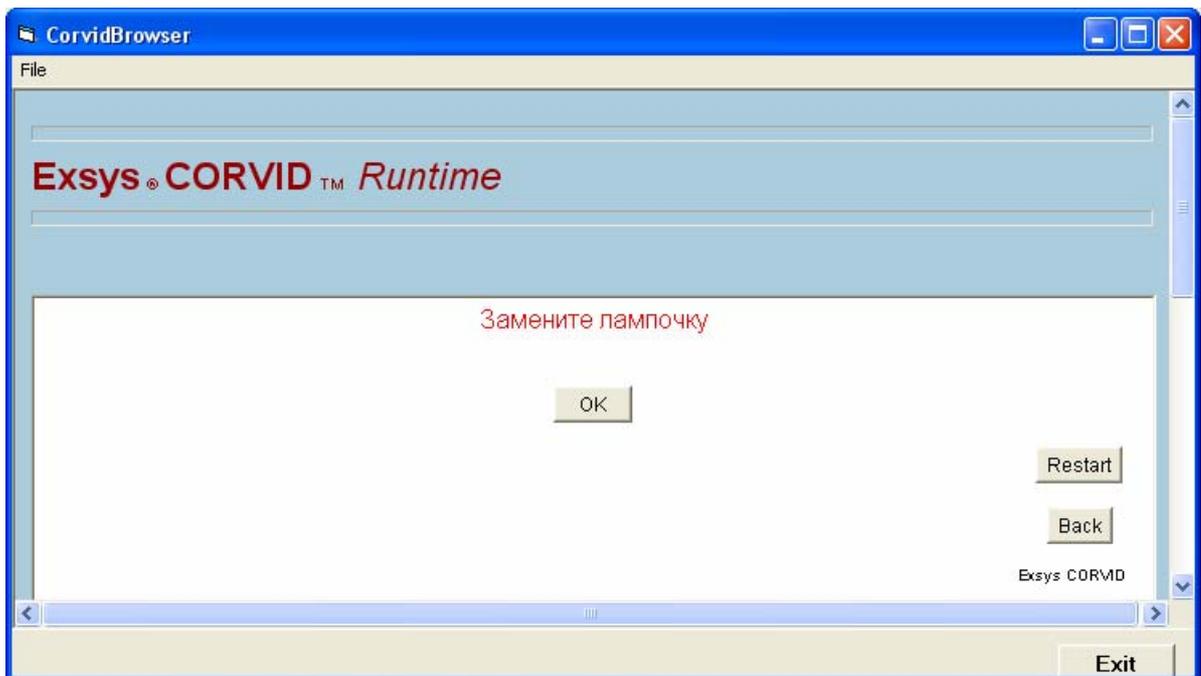
Первое окно, содержащее вопрос о работе освещения осталось без изменений. Выберите «внезапно перестал работать» и нажмите ОК.

Окно результатов отображается в соответствии с форматом, заданным в команде RESULTS: размер шрифта, центрирование и цвет.

Следующий шаг – это форматирование формы опроса пользователя.

Окна, в которых производится опрос пользователя, могут быть форматированы индивидуально или в группе, используя стиль, применяемый ко всем вопросам.

Сначала определим формат индивидуально. Нажмите кнопку для отображения окна Variable.



Выделите переменную BULB.

На закладке Ask With можно определить способ отображения вопросов к пользователю: виды используемых элементов управления, формат, шрифт, цвет и пр.

Для изменения формата вопроса выполните следующее:

1. Выберите List (список) в качестве используемого элемента управления.
2. Укажите способ расположения (Arrange) Same as Prompt, который расположит значения рядом с комментарием (первой фразой) вопроса, а не под ним.
3. Нажмите кнопку Edit, расположенную напротив метки Prompt Format и измените размер шрифта на 14 pt.

Для просмотра того, как будет выглядеть отображения вопроса, нажмите кнопку Preview на закладке Ask With.

Для просмотра всех вопросов системы нажмите кнопку Preview All.

По нажатию какой-либо кнопки Preview создается временный командный блок для тестирования вопросов, связанных со значениями переменных. В независимости от даваемых ответов будут заданы все вопросы. Это удобный способ проверки вопросов, в особенности, когда используются верхние и нижние колонтитулы на страницах вопросов.

Как Вы видите, список возможных значений переменной расположен рядом с комментарием. Выберите какое-нибудь значение для завершения предварительного просмотра и возврату к редактированию.

В случае, когда требуется более сложный интерфейс ввода ответов пользователя, можно использовать графические карты – Image Maps. Эти карты используют файлы в формате JPG или GIF. На изображении задаются активные зоны – Hot Spots, и, когда пользователь щелкает мышкой на

активной зоне, соответствующей переменной присваивается значение, соответствующее ответу на данный вопрос.

Чтобы использовать активные зоны выделите имя переменной BULB и нажмите кнопку Edit Map. На экране появится окно редактора активных зон Hot Spot Editor.

Первым шагом следует загрузить файл изображения, предназначенного для ввода ответа на вопрос. Как уже было отмечено, файл может быть в формате JPG или GIF. Для выбора файла используйте кнопку Browse.

В данном случае используйте заранее подготовленный файл bulbimage.gif, который может использоваться для ввода ответа на вопрос системы.

Определение активной зоны заключается в выделении на изображении прямоугольной области и ассоциировании какого-либо значения переменной с данной областью. Во время работы системы, если пользователь щелкнет мышкой над выделенной областью, то будет присвоено значение ассоциированной переменной, или, если указана ссылка URL, то она будет открыта в новом окне.

Для выделения требуемой области на изображении просто нажмите и удерживайте левую кнопку мыши в верхнем левом углу области, затем переместите курсор мыши в нижний правый угол области и отпустите кнопку. В процессе выделения будет отображаться соответствующий прямоугольник. После отпускания мышки прямоугольник будет оставаться подсвеченным, а его углы – выделены маркерами, с помощью которых можно изменить пропорции и размеры активной области.

После того, как активная зона выделена, необходимо указать, что именно будет ассоциировано с этой областью. В данном случае требуется присвоить значение переменной BULB. Значение в данном случае должно быть «перестал_работать». Это выполняется в группе On Click.

Для изменения значения какой-либо области, просто щелкните на ней мышкой и выберите новое значение. С помощью группы Find можно найти все активные зоны, ассоциированные с указанными переменными и их значениями. В группе Coordinates отображаются координаты активной зоны. Там же можно изменить координаты путем ввода требуемых значений в соответствующие поля.

На следующем шаге необходимо указать активную зону для другого значения. Выделите на изображении область «продолжает работать» и укажите соответствующее значение переменной – «продолжает_работать».

Это все, что необходимо для определения графической карты, таким образом, нажмите кнопку Done.

Запустите систему для проверки работы активных зон. Первый вопрос будет задан с помощью графической карты. Обратите внимание, что при использовании графических карт кнопка ОК не выводится на экран.

Лабораторная работа 3: Усиление логики работы системы

Цель работы: усовершенствовать логический блок имеющейся экспертной системы

Шаг 1

В первой работе мы создали экспертную систему которая рекомендует поменять лампочку всякий раз когда гаснет свет. Но в реальном мире такая система не найдет применения. Если выключатель в комнате выключен система будет рекомендовать заменить все лампочки в комнате, а если вдруг отключат электричество, мы будем вынуждены заменить все лампочки в доме.

Ясно, что система нуждается в улучшении логики работы, для того чтобы она сумела обработать различные случаи отключения лампочек.

Что бы начать работу, откройте систему созданную на первом и втором уроке.

Шаг 2

Для того чтобы создать логику решения проблемы заключающейся в выключателе и отключении электричества в доме, нам нужно больше информации.

Мы должны знать, если погасла лампочка, погаснут ли другие лампочки в доме и комнате.

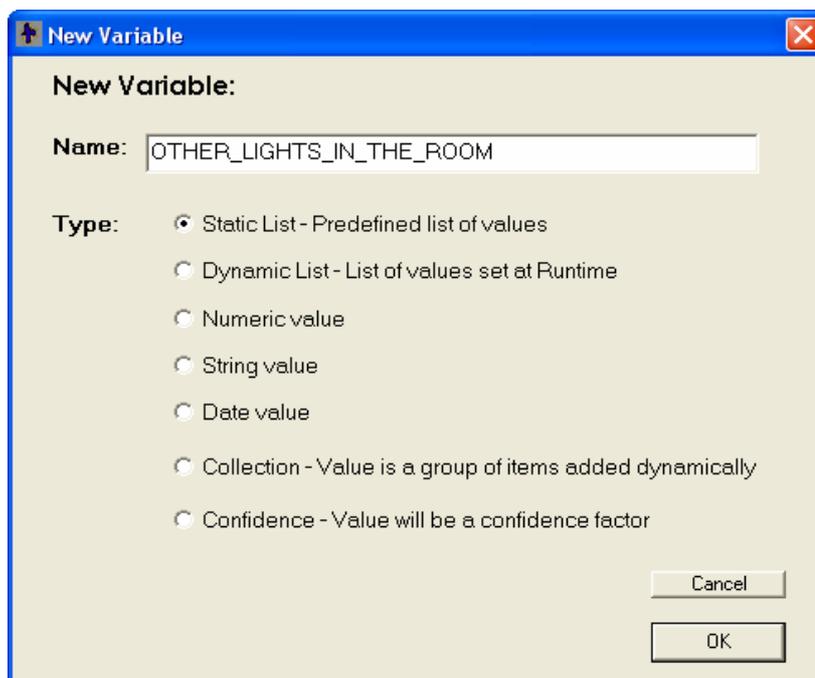
Для того, что бы запросить у пользователя эту информацию, добавим новую переменную типа **static list** (переменную со статическим списком)

Кликните на иконку окна переменных(VARIABLE WINDOW)  чтобы открыть окно добавки новых переменных.

В открывшемся окне кликните на кнопку NEW (Новая переменная).

Шаг 3

Новая переменная нужна чтобы спросить у пользователя, погасли ли другие лампочки в комнате. Назовем переменную “Other_lights_in_the_room” (Другие лампочки в комнате)



Имя вводится в окно правки Name (Имя). В имени можно использовать пробелы, но CORVID их конвертирует в символы подчеркивания.

В списке тип переменной следует установить значение **static list**

Кликните ОК чтобы добавить новую переменную.

Шаг 4

Теперь новая переменная находится в списке переменных.

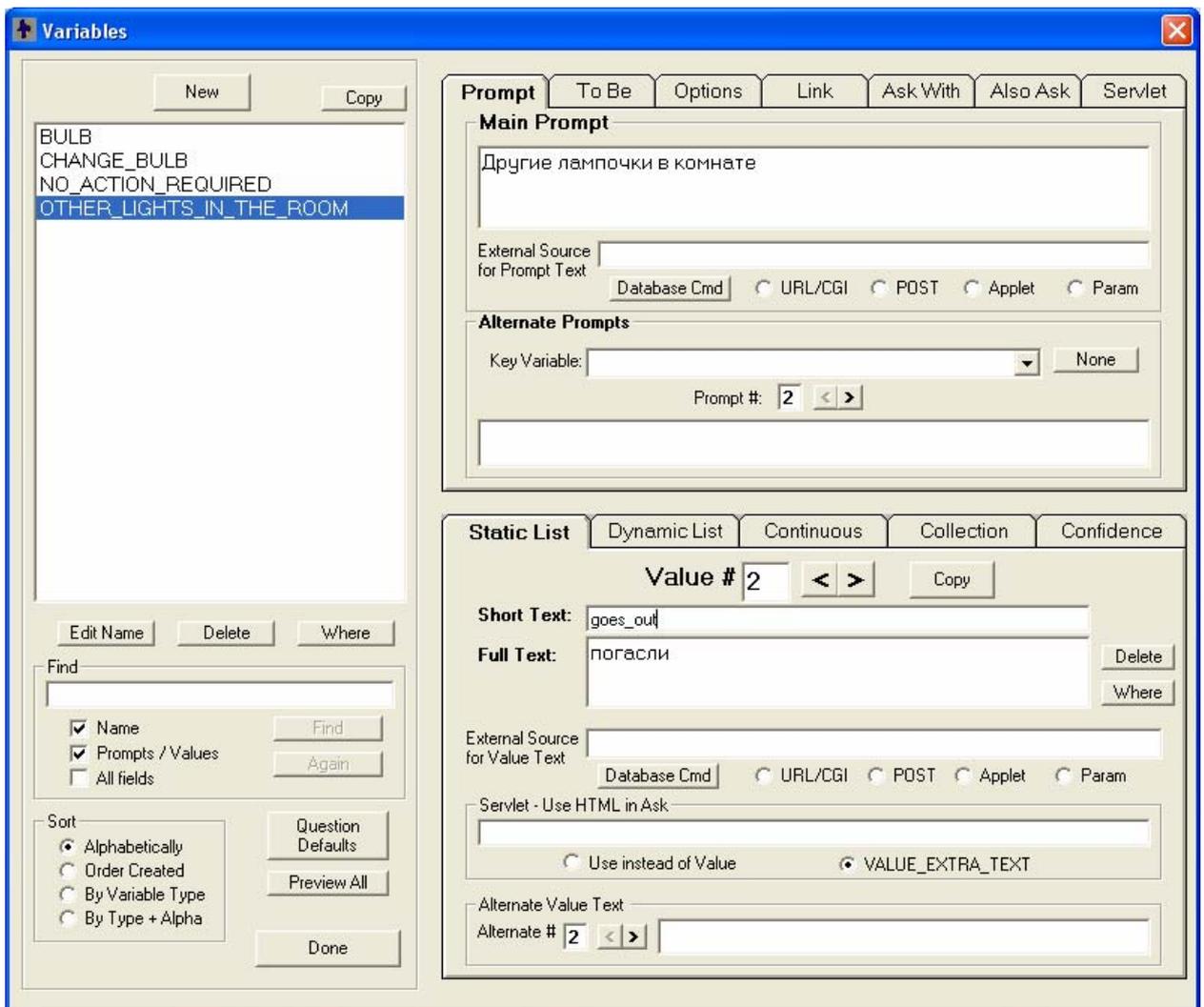
Измените текст подсказки на “Другие лампочки в комнате”, Введите в поле Full Text(полный текст) “Stay on”(продолжают гореть) – это первое значение переменной, затем кликните Copy(копировать) для того чтобы скопировать его в поле Short text(короткий текст).

Кликните на кнопку (>) чтобы перейти к следующему значению

Шаг 5

Теперь введите второе значение переменной - “go out” (погасли) в поле Full text, а затем скопируйте в поле Short text.

Переменная нуждается только в двух значениях, после того как вы ввели второе, перейдите на вкладку Options и в списке “Maximum Number of Values that can be Assigned”(Максимальное число значений которые могут быть присвоены переменной) выберете “Single value” (Одно значение). Это оградит пользователя от выбора обоих значений сразу. В вопросе будет использоваться переключатель, чтобы упростить ввод единственного значения.



Шаг 6

Сейчас мы добавим рекомендации для случая, когда проблема заключается в сломанном выключателе. Это делается путем добавления новой переменной типа **confidence** (Доверительной переменной)

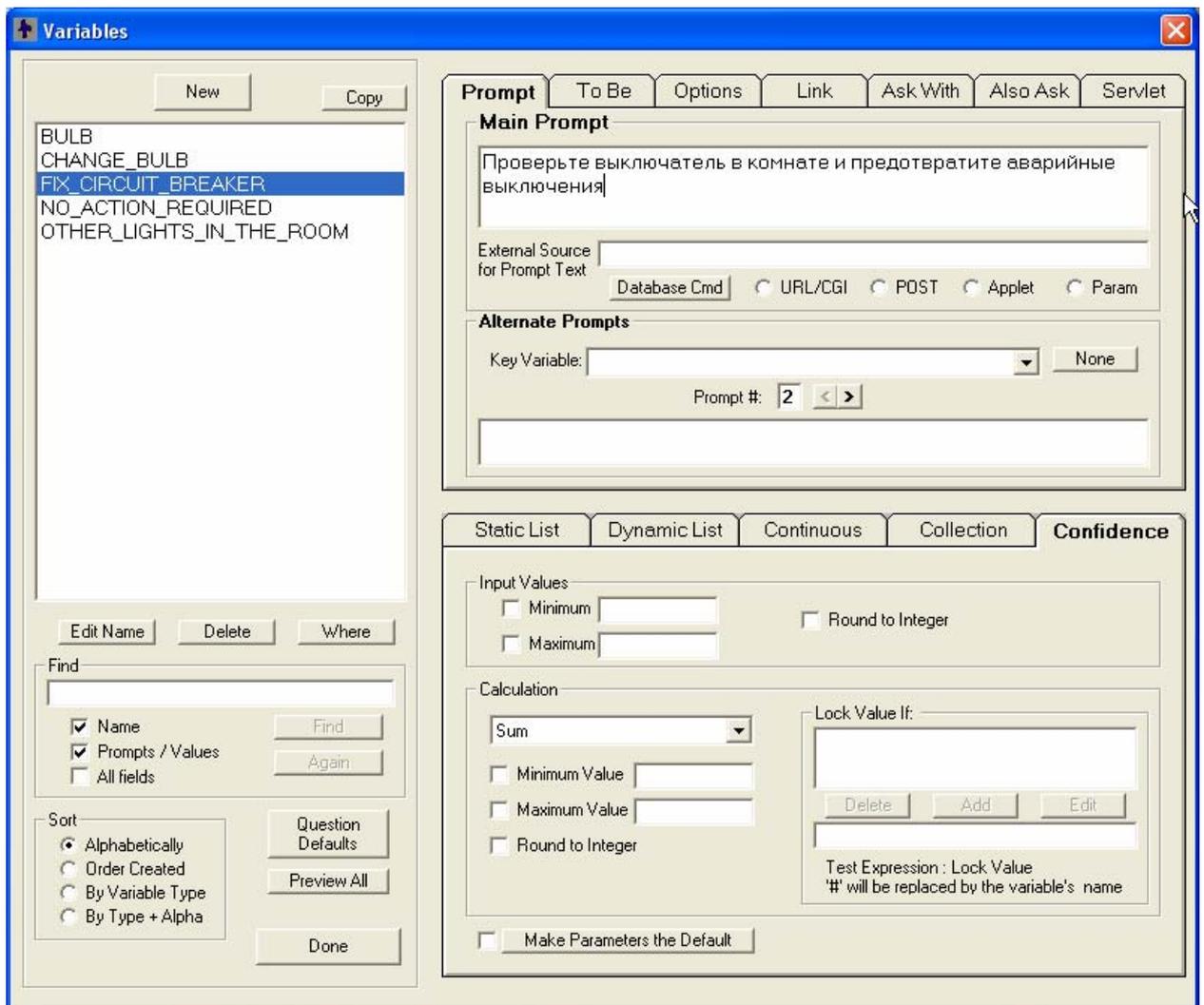
Нажмите на кнопку **NEW** чтобы добавить новую переменную, назовите её “Fix circuit breaker”(Проверка выключателя)

Тип переменной установите **confidence**

Нажмите **OK** чтобы добавить новую переменную.

Шаг 7

Переменная добавлена, текстовая подсказка, введенная в поле **prompt** (подсказка), будет рекомендацией появляющейся при выполнении необходимых условий. Измените строку подсказки на “Check the circuit breaker for the room and reset any tripped breakers” (Проверьте выключатель в комнате и предотвратите аварийные выключения.)



Шаг 8

Теперь нам нужна переменная которая будет спрашивать о состоянии других лампочек в доме. Эта переменная будет использоваться для подтверждения того, что произошло отключение электричества.

Добавьте новую переменную типа **static list**, назвав её “Other lights in the house” (Другие лампочки в доме), Добавьте для неё значения “Stay on”(продолжают гореть) и “Go out”(погасли). Установите для этой переменной выбор только одного значения (Single Value)

Шаг 9

Наконец, пора добавить последнюю переменную – рекомендацию в случае отключения электричества

Добавьте новую переменную типа **confidence** с именем “Call the power company” (Позвоните поставщику электроэнергии). А в строке подсказки напишите «Call the power company and report the problem» (Позвоните поставщику электроэнергии и сообщите о проблеме)

Шаг 10

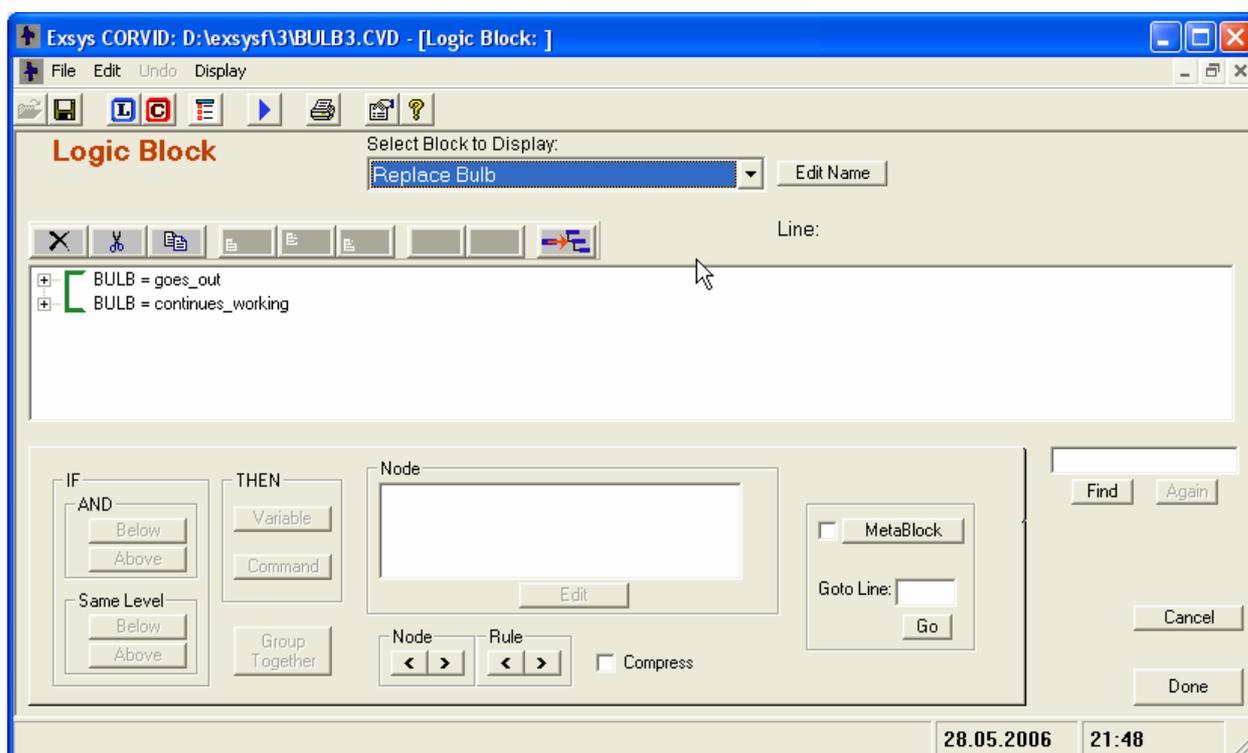
Вот все те переменные, которые нам нужны для усиления логики работы экспертной системы.

Нажмите DONE чтобы закрыть окно переменных.

Шаг 11

Теперь откроем логический блок исходной системы и будем его расширять.

Кликните на иконке логического блока, чтобы открыть окно логического блока



Выберете в выпадающем списке “replace bulb” (Замена лампочки). Разверните список операторов, кликнув на иконку раскрытия списка.

Нам нужно усилить логику которая определяет что делать в случае если лампочка погасла.

текущее правило таково:

IF (если) The bulb goes out (лампочка погасла) THEN (тогда) Change the bulb (замените лампочку)
--

теперь учитывая фактор других лампочек в комнате, создадим немного более общее правило.

```
IF
    The bulb goes out (лампочка погасла)

AND
    The other lights in the room stay on (другие лампочки в комнате
    продолжают гореть)

THEN
    Change the bulb (замените лампочку)
```

Для этого нужно добавить второе условие, добавленное с помощью оператора AND. Только если оба условия (IF и AND) будут истиной, весь блок IF будет истиной.

Для того чтобы добавить второе условие кликните на строчку “BULB = goes_out”, в окне просмотра правил (Rule View Window) появится условие IF.

Шаг 13

Теперь изменим текущее правило

```
IF
    The bulb goes out (лампочка погасла)
THEN
    Change the bulb (замените лампочку)
```

На новое

```
IF
    The bulb goes out (лампочка погасла)

AND
    The other lights in the room stay on (другие лампочки в комнате
    продолжают гореть)

THEN
    Change the bulb (замените лампочку)
```

Чтобы добавить новое условие IF ниже выбранного условия IF кликните на кнопку AND – Below (Добавить ниже) в нижней части окна в группе IF.

Шаг 14

На экране появится окно для добавки новых условий IF. Мы желаем использовать переменную “Other_lights_in_the_room”(другие лампочки в комнате). Кликните на неё чтобы выделить. Выделив переменную, мы сможем увидеть её значения во вкладке “Static list” в правой части окна.

Шаг 15

При добавлении узлов (nodes) IF, (их можно добавлять сразу несколько). Каждый из них будет формировать нашу ветку в логическом дереве. В большинстве случаев было бы неплохо включать каждое из возможных значений в ветку, если это делать, то логика сможет предусмотреть все значения вводимые пользователем.

Если выбранный узел имеет другой узел соединенный с ним узел(к таким узлам относятся, любой THEN узел или IF с добавленными с помощью оператора AND условиями). И мы добавляем новые составные узлы, то первый из узлов присоединяется к выбранному узлу, а остальные к нему снизу. Какие бы другие узлы логического дерева мы ни добавляли, они будут добавлены под выбранный узел и присоединены оператором AND, но не будут иметь присоединенных узлов логического дерева под ними. Это может звучать путано, но когда вы увидите пример, все станет понятней.

В нашем случае мы добавляем условие IF между существующими IF и THEN.

Настоящее правило таково:

IF	The bulb goes out (лампочка погасла)
THEN	Change the bulb (замените лампочку)

Мы хотим сделать:

IF	The bulb goes out (лампочка погасла)
THEN	Change the bulb (замените лампочку)

На новое

IF	The bulb goes out (лампочка погасла)
AND	

The other lights in the room stay on (другие лампочки в комнате продолжают гореть)

THEN

Change the bulb (замените лампочку)

Также хотим добавить новое правило:

IF

The bulb goes out (лампочка погасла)

AND

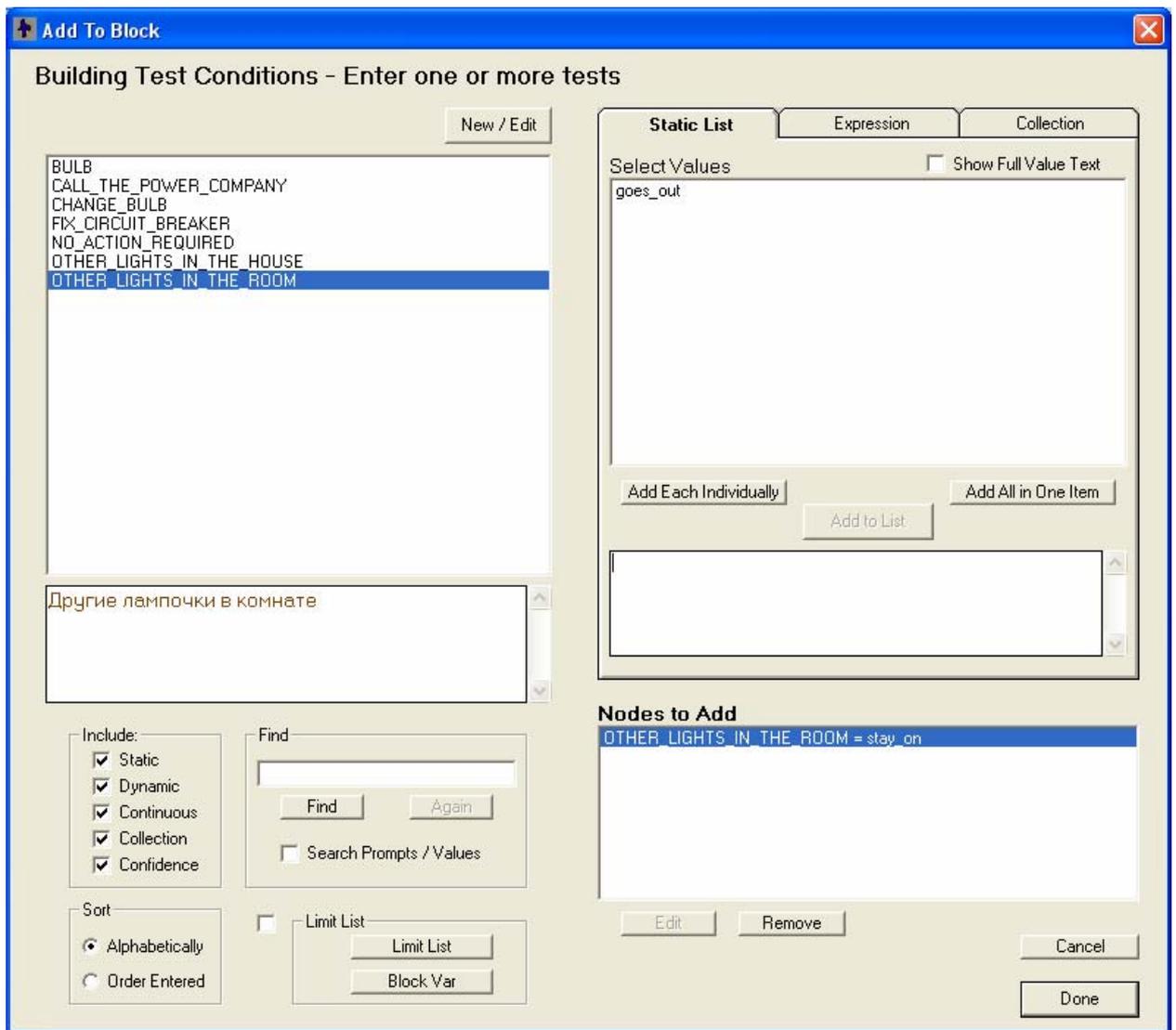
The other lights in the room goes out (другие лампочки в комнате продолжают гореть)

Шаг 16

The other lights in the room go out (Другие лампочки в комнате погасли)
– узел, который мы хотим добавить в существующее правило.

Мы уже имеем список значений переменной. А теперь мы можем добавить эти значения в список (Nodes to ad list).

Кликните на значении Stay on, чтобы выбрать его. Полный текст замечания отобразится в среднем окне. Кликните на кнопке ADD TO LIST (Добавить в список) чтобы добавить его в список добавки узлов.



Мы добавим два узла. И Важно чтобы узел “stay on” (продолжают гореть) был первым в списке. Именно этот узел будет вставлен в существующее правило под выбранным узлом.

Шаг 17

Сейчас мы добавим второй узел, он будет также добавлен с помощью оператора AND к выбранному правилу, но не будет иметь под собой оператора THEN.

Этот узел был добавлен чтобы начать создавать новое правило, которое будет завершено нами позднее.

Существует несколько путей, чтобы добавить этот узел. Можно повторить шаги, которые мы только что сделали и кликнуть на “go_out”, затем кликнуть на кнопку ADD TO LIST, чтобы добавить узел в список. Второй путь это воспользоваться специальными кнопками: “ADD EACH INDIVIDUALLY” (Добавить каждую по отдельности) и “ADD ALL IN ONE ITEM”(Добавить все вместе). “ADD EACH INDIVIDUALLY” добавляет оставшиеся не присвоенные значения в список, по одному на узел. А кнопка

“ADD ALL IN ONE ITEM” Добавляет все не присвоенные значения в один узел. Поскольку мы имеем только одно оставшееся значение, обе эти кнопки будут выполнять одно и тоже действие.

Нажмите на кнопку “ADD EACH INDIVIDUALLY” чтобы добавить значение “go_out” в список как узел.

Мы сейчас создали два узла, нажмите DONE чтобы добавить их в логический блок.

Шаг 18

Новые узлы сейчас были добавлены в логический блок. Узел IF “Other_lights_in_the_room=stay_on”(Другие лампочки в комнате = продолжают гореть) появляется под узлом “BULB=goes_out” (лампочка погасла). Такое расположение идентифицирует эти узлы как единое условие. Узел “[Change bulb]=10” является узлом THEN который присоединен с низу к узлу “Other_lights_in_the_room=goes_out” и его идентификация будет зависеть от того будет ли истиной этот узел и соединенные с ним узлы.

Второй новый узел, добавленный нами, также соединен с узлом “BULB=goes_out” и находится под ним, но он ещё не имеет ни одного узла THEN (это показывает красная скобка перед ним). Нижний узел “BULB=continues_working” не изменился.

Узел “Other_lights_in_the_room=go_out” сейчас выбран, потому что он был последним узлом добавленным в список “Nodes to Add” (Добавляемые узлы). Пока он выбран в окне “Rule View”(просмотр правил) можно увидеть правило соответствующее данному узлу.

Логический блок показывает структуру логики, но чтобы видеть действующее правило, которое мы только что создали кликните на узел THEN “Change Bulb=10”. В окне “Rule View” появится новое правило. Также цветом выделятся все узлы участвующие в построении этого правила. В нашем случае они следуют друг за другом, но в более сложных системах могут быть разделены другими узлами.

Шаг 19

Следующий шаг – завершить правило которое мы начали с добавления узла “Other_lights_in_the_room=go_out”

Кликните на этот узел и в окне “Rule View” вы увидите полный текст.

Это правило пока ещё не содержит узлов THEN.

Шаг 20

Мы начали создавать правило обрабатывающее ситуацию, когда лампочка погасла, и другие лампочки погасли вместе с ней. Чтобы дать пользователю указания – что делать, Нам нужно знать: погасли все лампочки

в доме (отключилось электричество), или только в одной комнате (поломался выключатель).

Добавим два новых условия IF с помощью оператора AND под строчкой “Other lights in the room=go out” для того чтобы создать два новых правила. “Other lights in the room=go out” уже выбрано. Нажмите на кнопку “AND-BELOW ” в контрольной группе IF

Шаг 21

Надо создать два оператора для проверки состояния других лампочек в доме. Поскольку под выделенным оператором не имеется других, нам не важно в каком порядке мы будем их добавлять.

Возьмем первую переменную “Other lights in the house”, кликните чтобы выбрать её. Во вкладке Static list появятся её значения.

Чтобы добавить по одному оператору на каждое значение нажмите на кнопку “Add each individuality” Это добавит два оператора в список “nodes to add”. Это все что нам было нужно, нажимайте на кнопку done.

Шаг 22

Новые операторы добавлены в логический блок. Чтобы проверить правило которое будет ими создано посмотрите на окно Rule view.

Поскольку “Other lights in the house = go out” был последним добавленным оператором, Он выделен в логическом блоке и показывается в окне Rule view

чтобы посмотреть предыдущее правило в логическом блоке, нажмите на кнопку “Rule <” Это приведет к выводу на экран предыдущего правила.

В этом правиле пока еще нет не одного оператора THEN.

Шаг 23

Основываясь на условиях IF данного правила, соответствующая рекомендация будет проверять выключатель в комнате.

Эту рекомендацию мы добавим в логический блок как оператор then.

Для начала закройте окно Rule View нажав на кнопку ОК. затем кликните на оператор «Other lights in the room = stay on» чтобы выделить его и сделать доступными кнопки внизу экрана.

Шаг 24

Оператор Then создается путем установления значений переменной типа confidence ”Fix circuit breaker”

Необходимо помнить что все операторы THEN, являются назначениями значения переменной, и в особых случаях командами.

Чтобы добавить оператор Then, нажмите на кнопку Variable в группе THEN.

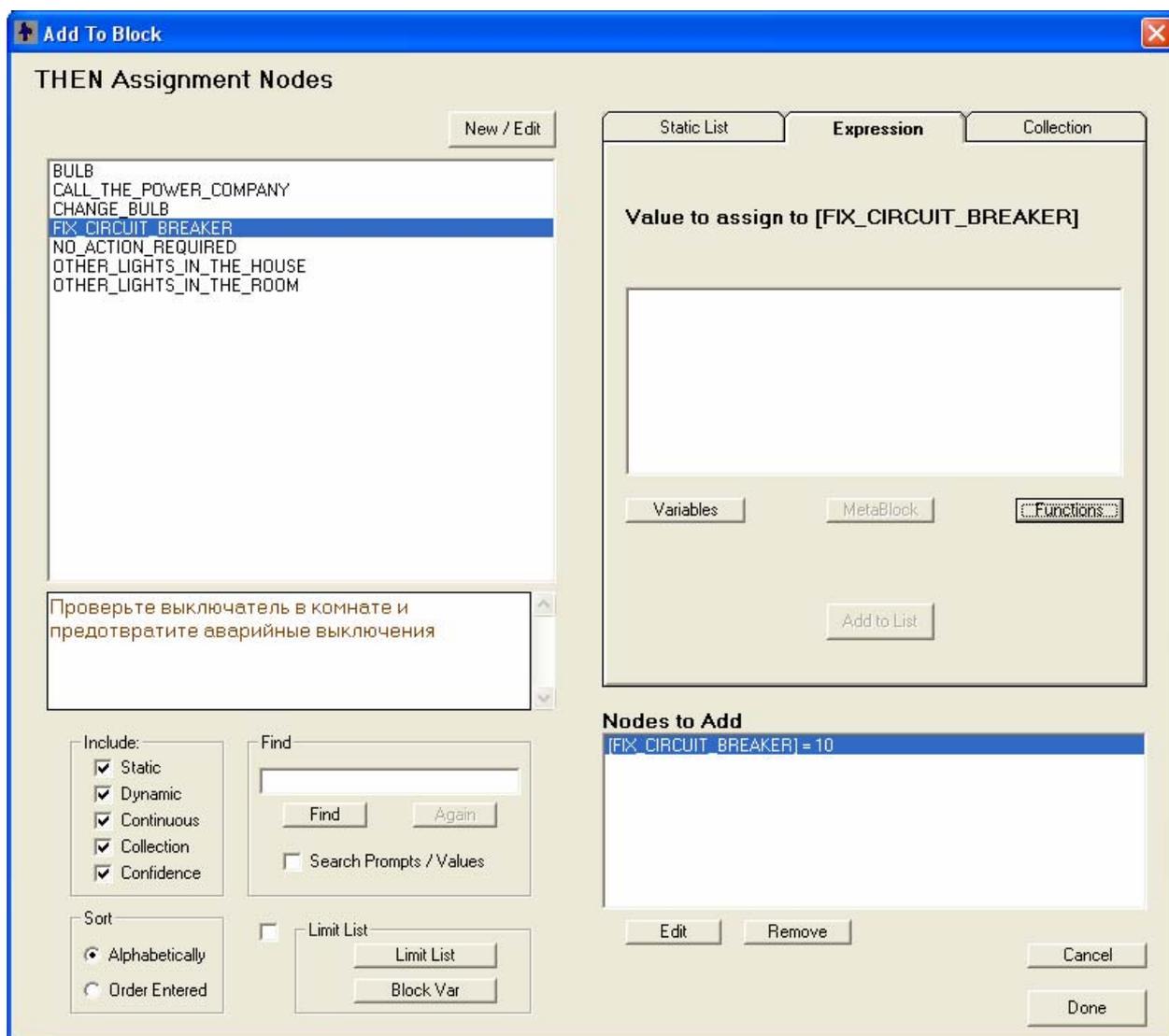
Шаг 25

На экране появилось окно добавки переменных. Теперь присвоим значения переменной.

Необходимо помнить что в отличие от IF узлов которые являются проверками и могут принимать значения только ИСТИНА или ЛОЖЬ, THEN узлы присваивают переменной значения.

Для начала выберем переменную для того чтобы присвоить ей значение. Кликните на переменную «Fix Circuit breaker» чтобы выбрать её. В открывшейся вкладке expression (выражение) нужно дописать часть выражения «[Fix circuit breaker]=»

Мы используем значение равное 10 для указания на выбранную рекомендацию. поэтому напишите 10 и нажмите на кнопку Add to list.



Один then узел добавлен, нажмите на кнопку done.

Шаг 26

Новый Узел Then Добавлен в логический блок.

Теперь завершим создание логического блока добавив в него подобный узел для ситуации, когда все лампочки в доме погасли. Кликните на узел «Other lights in the house = go out» чтобы выделить его, и добавьте узел присваивающий значение 10 доверительной переменной “Call power company”.

Шаг 27

Чтобы проверить созданное нами правило кликните на узел «Call the power company = 10» удерживая при этом правую клавишу shift. В окне просмотрщика правил (Rule view) мы увидим правило.

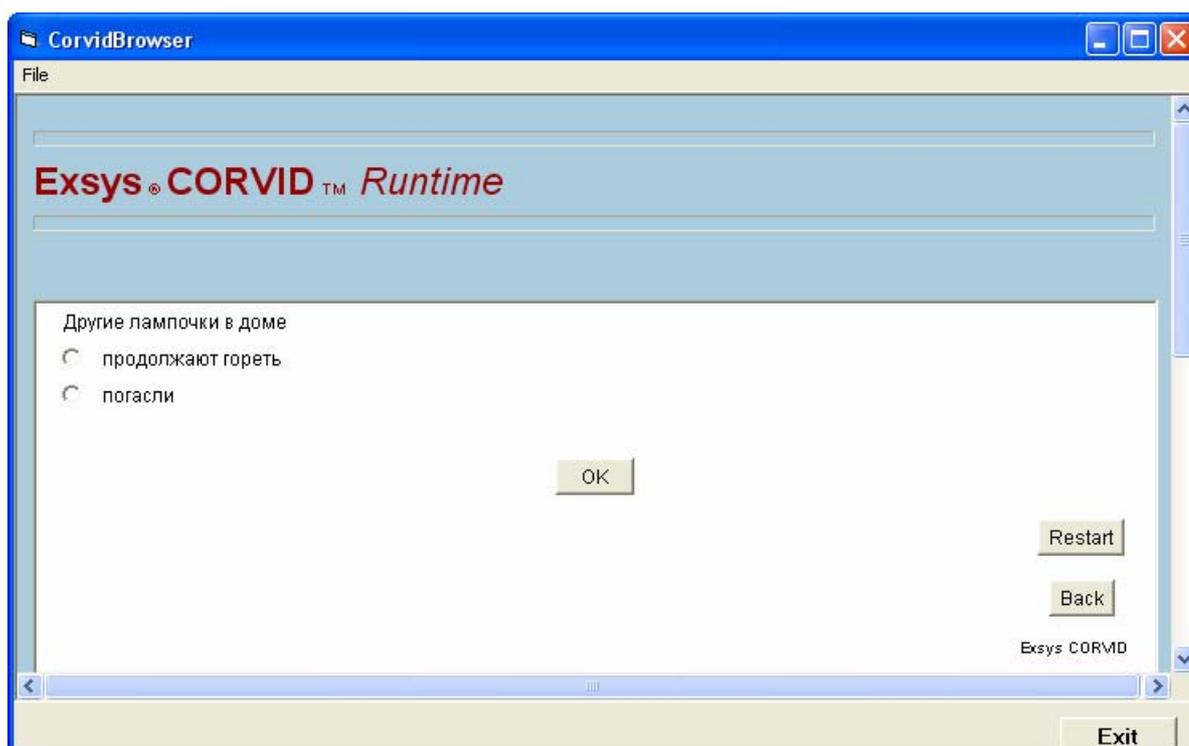
Шаг 28

Теперь логический блок имеет все дополнительные правила которые мы хотели добавить

Закройте логический блок нажатием на кнопку DONE

Шаг 29

Теперь мы можем запустить усиленную систему. Кликните на иконку запуска. Первый вопрос будет таким же как и прежде. Но если мы ответим что лампочка не горит, система спросит нас об остальных лампочках в комнате и доме.



Лабораторная работа 4: Обратная связь

Цель работы: Изучить обратную связь в EXYS CORVID

Шаг 1

Обратная связь – Одно из мощнейших средств в EXYS CORVID

Обратная связь позволяет разложить сложную проблему на маленькие, легко определяемые сегменты, которые автоматически используются системой когда в этом возникает потребность.

Хоть Обратная связь и является очень мощным средством, она довольно проста в использовании. Если системе нужно знать значение переменной, и вы подготовили правило которое разрешает CORVIDу устанавливать значения для этой переменной, он автоматически достроит до конца правило, установив значение переменной. Все что вам нужно делать это добавлять правила установки значений переменной используемой в другом IF условии правила.

Правила Обратной связи могут быть установлены везде где в них есть необходимость. CORVID найдет и использует правила которые вы предусмотрели.

Шаг 2

Например, в предыдущей работе мы создали правило которое включало в себя условие IF:

The other lights in the house go out (другие лампочки в доме погасли)

Чтобы определить правило системе нужно знать значения переменной [Other_lights_in_the_house]

До сих пор, когда системе были нужны данные, она запрашивала их у пользователя. Но если мы подготовим правило, разрешающее установку значения переменной, то CORVID попытается сам закончить правило.

Допустим мы добавили в систему правило:

IF
You hear the radio playing in another room (Ты слышишь радио в соседней комнате)
THEN
The lights in the rest of the house stay on (Другие лампочки в доме горят)

Вместо того чтобы спросить горят ли другие лампочки в комнате, система спросит слышит ли он радио в соседней комнате, И если пользователь ответит что слышит, CORVID может установить значение (другие лампочки в комнате) и не спрашивать пользователя.

Если пользователь ответит что он не слышит радио, то вызванное правило не сработает и не установит значение для переменной [Other_lights_in_the_house]. В этом случае CORVID будет искать другие правила, чтобы установить значение этой переменной, или прямо запросит его у пользователя.

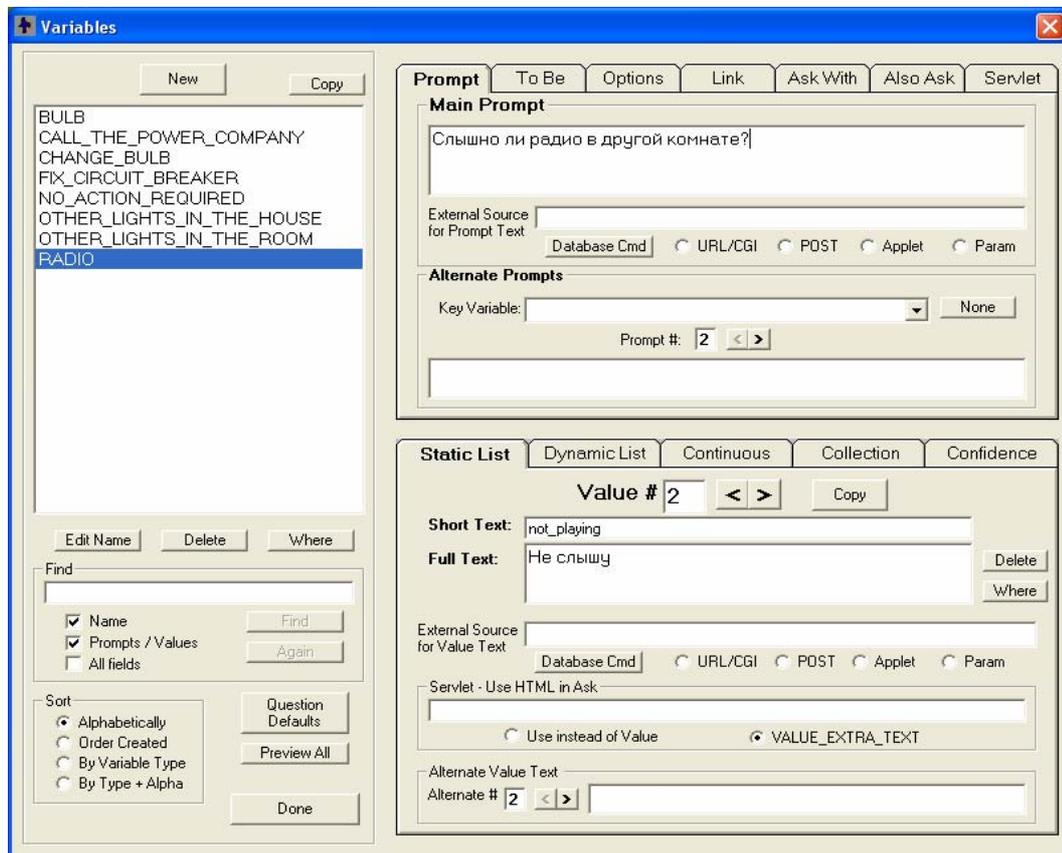
Заметка: CORVID вызовет любое правило которое сможет предоставить значение переменной, не только то которое сможет установить значение истина.

Шаг 3

Правило которое будет вызвано через обратную связь может быть добавлено в любом месте экспертной системы. Иногда удобно установить его в отдельный логический блок, что мы и сделаем. Создадим новый логический блок с именем “RADIO”. Кликните на иконку логического блока, чтобы открыть окно нового логического блока, и измените его имя на RADIO

Шаг 4

Нам понадобится новая переменная типа STATIC LIST. Назовем её RADIO в строке подсказки напишем «Радио в другой комнате» (THE RADIO IN THE OTHER ROOM) и значения «СЛЫШУ»(Can be heard playing) «Не СЛЫШУ»(Is not playing). Эта переменная будет использована в правиле вызываемом через обратную связь.



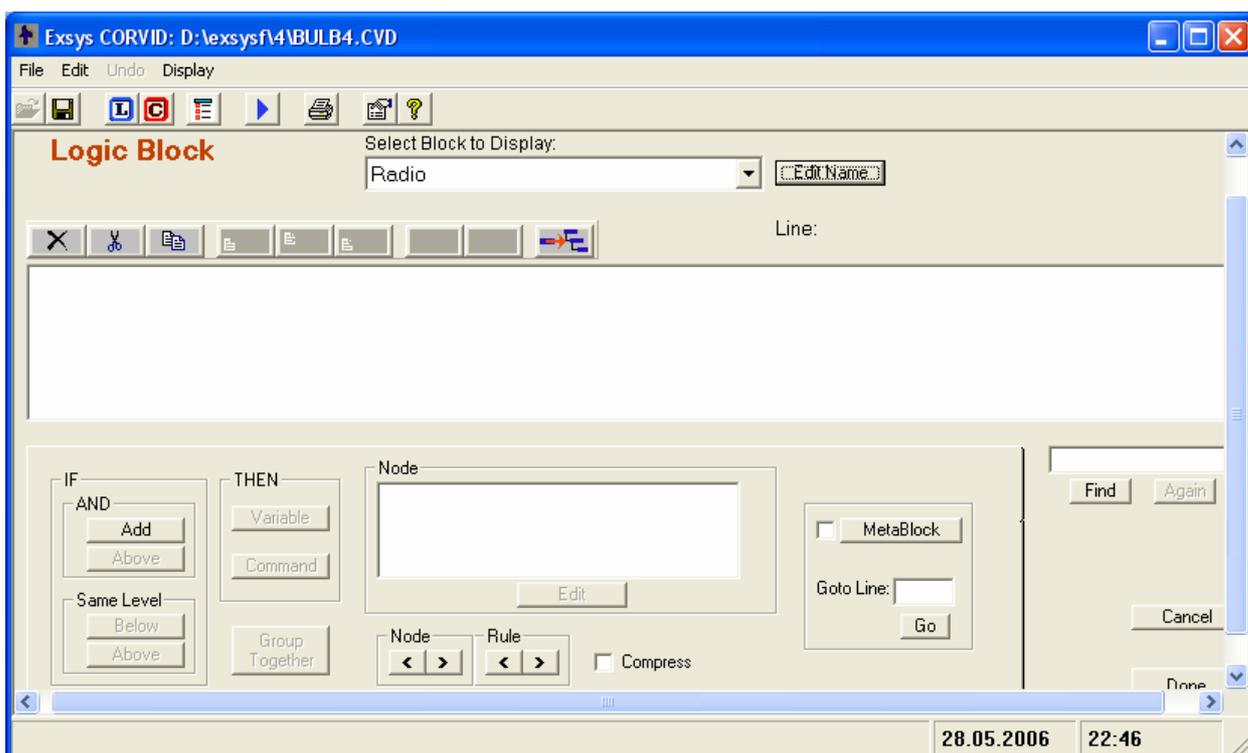
Шаг 5

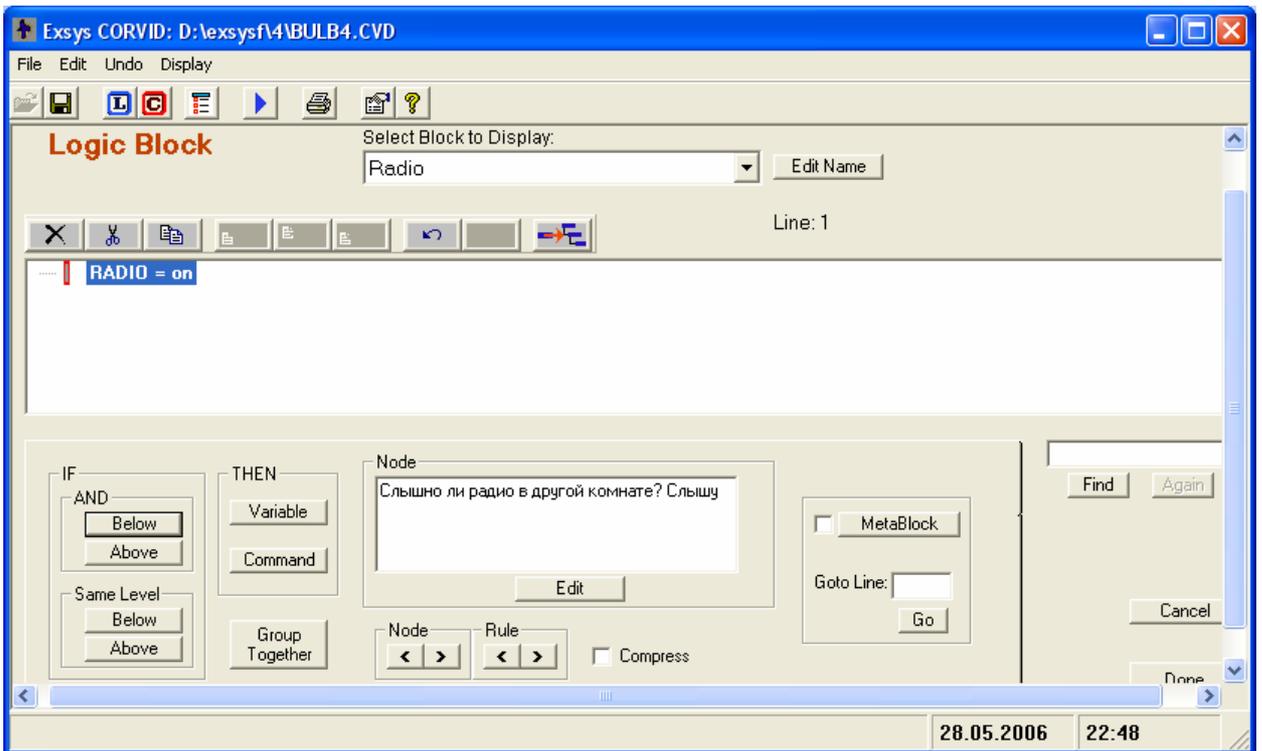
Добавьте узел в логический блок нажав на кнопку ADD в окне LOGIK BLOCK.

Шаг 6

Выберете переменную RADIO, на дисплее появятся её значения on и off (выключено). в нашем случае мы хотим создать правило только для одного значения on(включено). Если радио включено то мы знаем что электричество есть, а если выключено мы не знаем ничего (может быть его сегодня не включили или оно просто замолкло) поэтому нам не нужно правило для значения off(выключено).

Выберете значение on, добавьте его в список, затем нажмите done чтобы выйти. В логический блок будет добавлен один узел.





Шаг 7

Это простое правило и все что осталось сделать, это добавить узел then. кликните на кнопку variable в группе then.

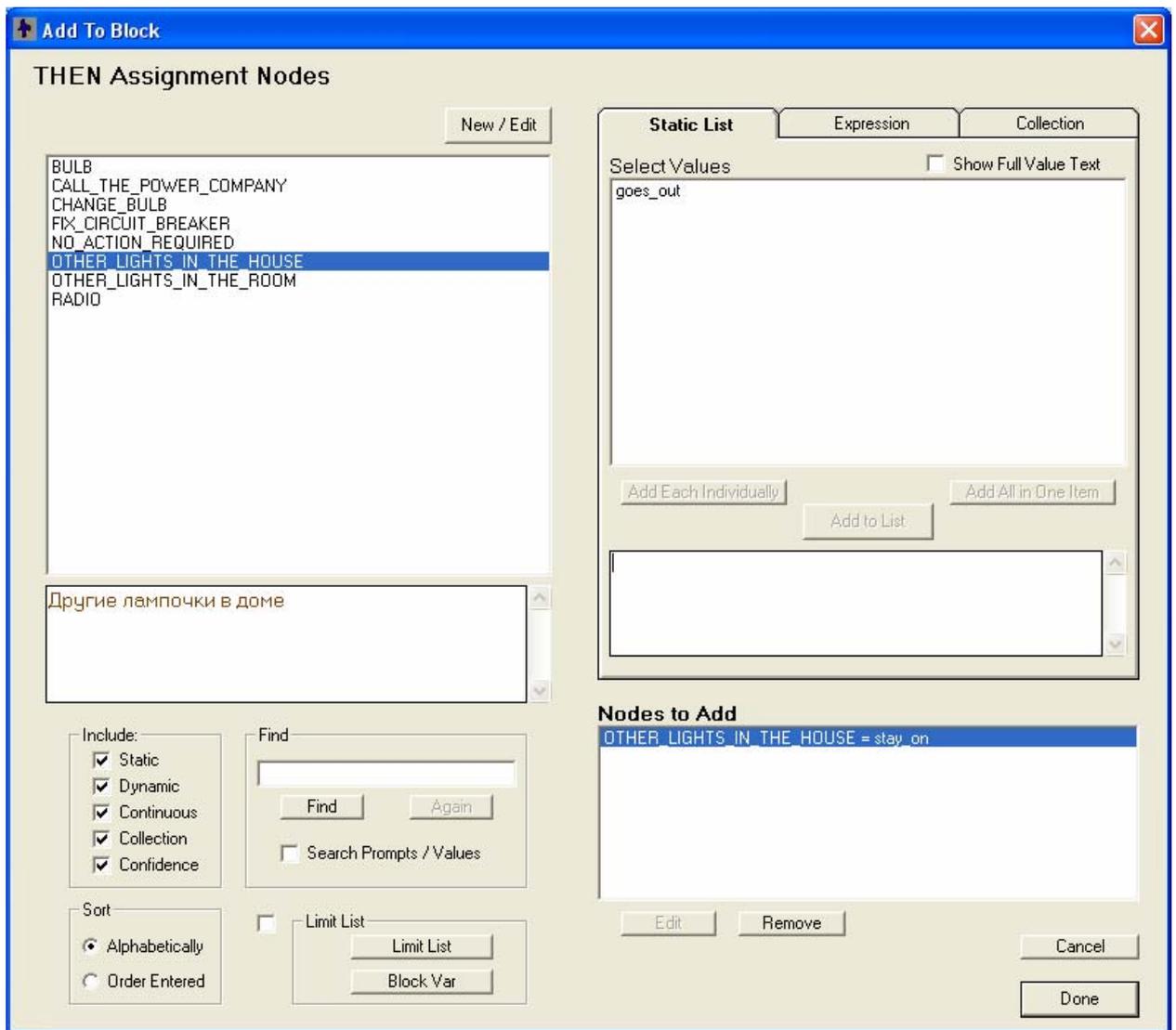
Шаг 8

Сейчас мы присвоим значение переменной типа static list. Обратная связь автоматически найдет правило, когда значение переменной понадобится системе. Правило будет проверено, и если оно сработает, то значение переменной будет установлено.

Шаг 9

Значение будем присваивать переменной “Other_lights_in_the_house”. Кликните на неё. На экране появится список значений, выберите из него stay_on и нажмите кнопку add to list.

Поскольку нам нужно присвоить значение только этой переменной нажмите done.



Теперь логический блок имеет одно правило, которое при необходимости может получить значение переменной “Other_lights_in_the_house”.

Шаг 10

Теперь запустим систему, первые два вопроса такие же как и раньше. Ответьте что лампочка погасла вместе со всеми лампочками в комнате.

Шаг 11

Теперь вместо того чтобы спросить горят ли другие лампочки в доме, система спрашивает слышим ли мы радио. Выберете что радио играет в другой комнате.

Шаг 12

Система вывела на экран своё заключение. Информация о свете в других комнатах была получена из нового правила.

Шаг 13

Если на вопрос о радио ответить что мы его не слышим, то система не сможет получить информацию свете в другой комнате. И так как других правил способных получить эту информацию нет, система спросит пользователя прямо.

Лабораторная работа 5: Числовые переменные и [[]] подстановки

Переменные являются основной единицей построения экспертных систем в EXYS CORVID.

До сих пор наша экспертная система использовала только два типа переменных:

1. Переменные со статическим списком
2. Доверительные переменные

Эта работа включает в себя:

1. Знакомство с числовыми переменными, которые используются для построения алгебраических выражений.
2. Дополнительные сведения о доверительных переменных.
3. Использование [[]] для включения значения переменной в текст.

Допустим мы решили улучшить нашу систему, чтобы она рекомендовала нам мощность сменной лампочки. (В целях экономии энергии мы решили не использовать лампочки мощностью более 75 ватт). Так правила будут использоваться следующим образом, если перегоревшая лампочка обладала мощностью менее 75 ватт, то её следует заменить на лампочку такой же мощности, а если мощность сгоревшей лампочки более 75 ватт то она заменяется на лампочку с номиналом 75 ватт. Мы собираемся менять лампочку толь в случае её выхода из строя, так что и правила должны инициироваться только в этом случае.

<p>IF Лампочка должна быть заменена AND текущая мощность больше 75 ватт. THEN Заменить лампочкой 75 ватт.</p> <p>IF Лампочка должна быть заменена AND текущая мощность меньше 75 ватт. THEN Заменить лампочкой той же мощности.</p>

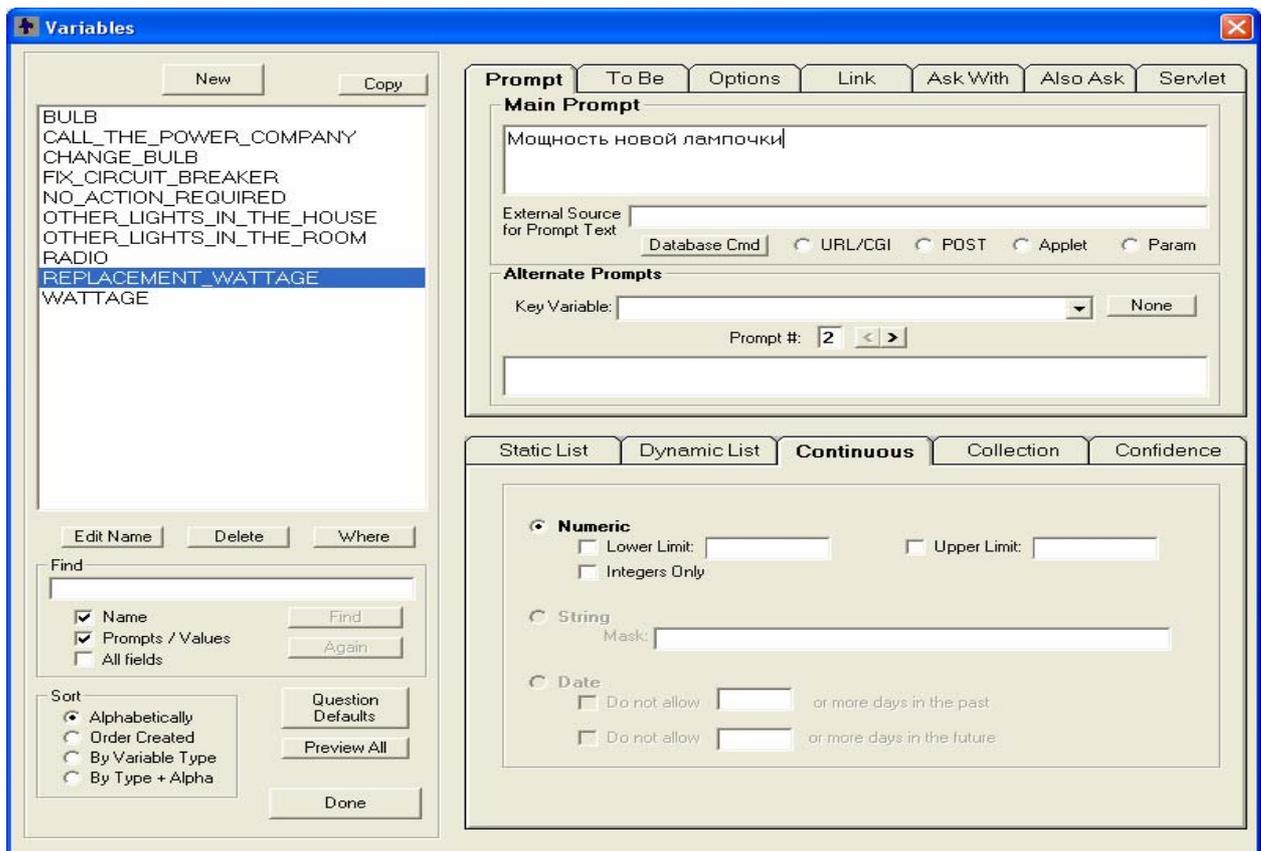
Шаг 1

Сначала добавим новую переменную. Это может быть переменная со статический список, которая просто спрашивает, превышает ли мощность лампочки 75 ватт, но для этого примера, создадим числовую переменную и назовем её WATTAGE (МОЩНОСТЬ), которая будет иметь значение точной

величины мощности лампочки. Переменная WATTAGE отображает закладку "Continuous". Она используется для переменных имеющих числовой диапазон, включая числовые, строчные и переменные даты. Пределы могут быть установлены в приемлемых входных величинах, но в нашей системе это не используется. В строке подсказки напишите: "Мощность сгоревшей лампочки"

Также добавьте вторую числовую переменную для мощности заменяемой лампочки. Система могла бы быть построена и без этой переменной, но она будет использована, чтобы показать некоторые интересные особенности CORVID.

The screenshot shows the 'Variables' configuration window. The left pane lists variables, with 'WATTAGE' selected. The right pane shows the configuration for the selected variable, including a 'Main Prompt' field containing the Russian text 'Мощность сгоревшей лампочки'. The 'Continuous' tab is selected, showing options for numeric, string, and date types.



Шаг 2

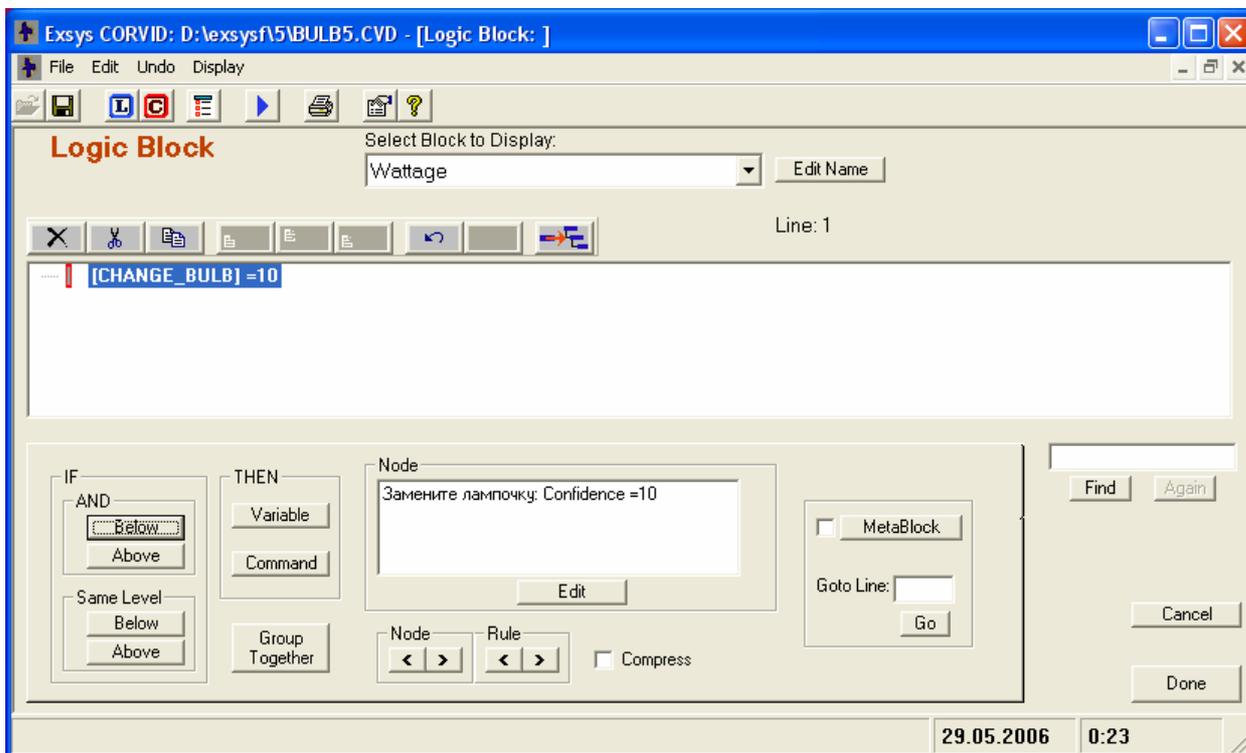
Добавьте новый логический блок, для определения мощности сменной лампочки. Назовите его "Wattage"(мощность).

Кликните кнопку "Add" в IF / AND группе, чтобы добавить первый IF узел.

Шаг 3

Мощность заменяемой лампочки представляет интерес только если лампочка должна быть заменена. Но чтобы правило существовало только для ситуации, когда лампочка должна быть заменена, используйте доверительную переменную "Change_bulb" для проверки выражения. Доверительные переменные могут иметь назначенные величины, которые комбинируются, чтобы произвести конечную величину. Эта конечная величина может быть использована в выражениях как числовая переменная. В нашем случае, переменной "Change_bulb" назначена величина 10, если лампочка должна быть заменена, так что добавляем тестовое выражение "[Change_bulb] = 10". Выберите переменную "Change_bulb" в списке. Это скопирует имя переменной в построитель выражения. Затем завершите выражение в окне построителя выражения.

Примечание: Когда переменная использована в выражении она должна быть заключена в квадратных скобках []. Этот IF узел будет истиной только если переменной доверия [Change_build] назначена величина 10.



Шаг 4

Поскольку эти правила должны срабатывать только если лампочка должна быть заменена, нужен только единичный IF узел. Если доверительная переменная [Change_bulb] имеет любую другую величину отличную от 10, ничего не должно быть сделано в этой секции.

Кликните кнопку " Below" в группе "IF / AND" , чтобы добавить узел, который будет определять превышает ли мощность лампочки 75 ватт.

Шаг 5

Новые узлы должны быть добавлены для того, чтобы проверять, превышает ли мощность заменяемой лампочки 75 ватт. Для этого они, используют переменную [WATTAGE], которая добавлялась раньше.

Выберете переменную WATTAGE чтобы копировать её в редактор выражений. завершите выражение "[WATTAGE]>75" и кликните на кнопку "Add to list"

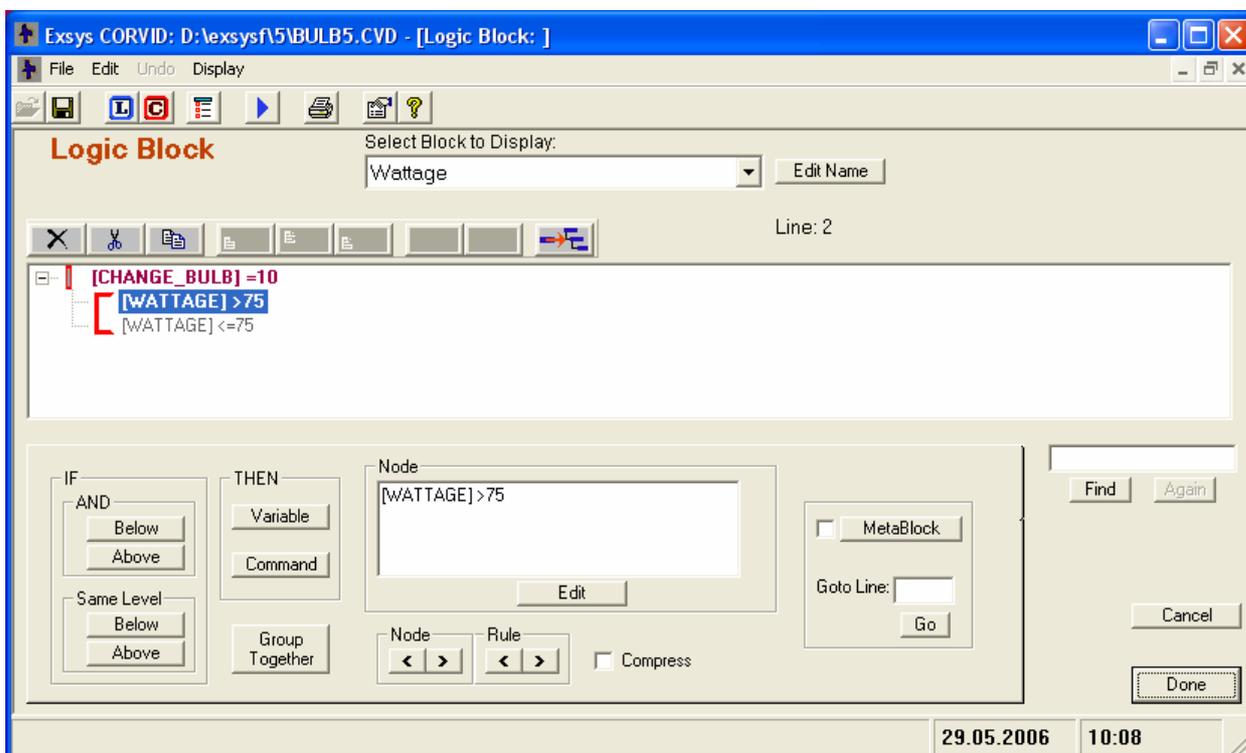
Так как правила нужно также рассматривать на случай где мощность не больше, чем 75 ватт, добавьте второй узел. Выберете переменную "WATTAGE" снова, чтобы копировать ее в поле редактирования выражений

и завершите второе выражение "[WATTAGE] <= 75" и нажмите кнопку Add to List.

Список Nodes to Add теперь имеет 2 узла, которые покрывают все возможные величины переменной [WATTAGE], так что нажмите кнопку DONE.

Шаг 6

При мощности сгоревшей лампочки больше чем 75 ватт, заменяемая лампочка должна быть 75 ватт. Добавьте THEN узел, чтобы установить величину переменной [Replacement_wattage] на 75. При выделенном узле "[WATTAGE] > 75", нажмите на кнопку Then\Variable

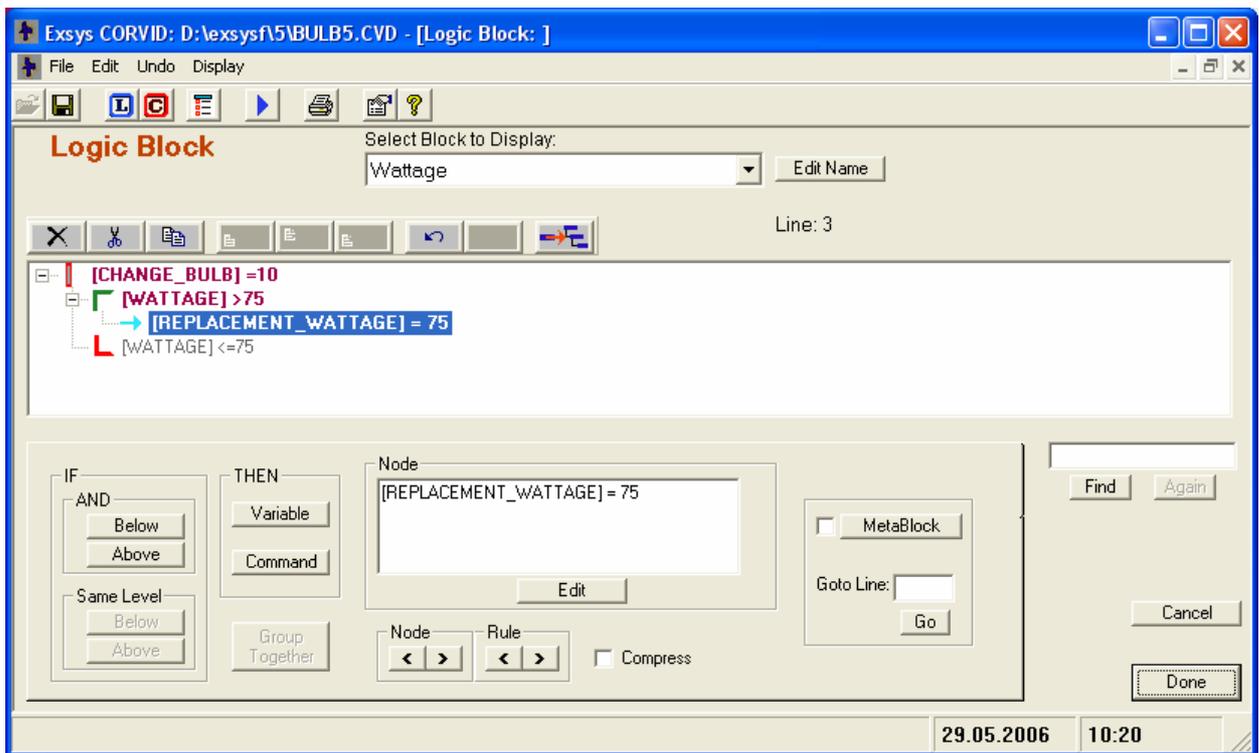


Шаг 7

В открывшемся окне присваиваем значение переменной [Replacement_wattage], окно построено подобно окну создания IF узлов.

Для того чтобы добавить THEN узел, выберите переменную, щелкнув на ней. Произойдет копирование переменной и знака "=" в окно редактирования выражений. Введите величину или выражение, которые должны быть присвоены переменной (в нашем случае это 75). Добавьте узел к списку и щелкните Done.

Логический Блок теперь имеет правило, чтобы устанавливать номинал для заменяемой лампочки когда мощность сгоревшей лампочки - выше 75.



Шаг 8

Теперь добавьте правило для случая, где мощность сгоревшей лампочки – 75 ватт или менее. В этом случае, заменяемая лампочка должна быть такой же мощности.

Щелкните на узле "[WATTAGE] <= 75", чтобы выбрать его, затем нажмите на кнопку Then\Variable

На этот раз вместо введения числа, величина берется из переменной [WATTAGE], которая содержит мощность сгоревшей лампочки. Вы могли просто набрать "[WATTAGE]" после знака "=", но легче получить переменную из списка. Установите курсор где вы хотите добавить переменную и щелкните на кнопке Variables, чтобы отобразить список переменных в системе.

Выберите переменную, чтобы добавить её к выражению. Когда переменная выбрана, вы можете дополнительно выбрать любое из свойств. В нашем случае, просто выбираете переменную "WATTAGE " и нажимаете ОК.

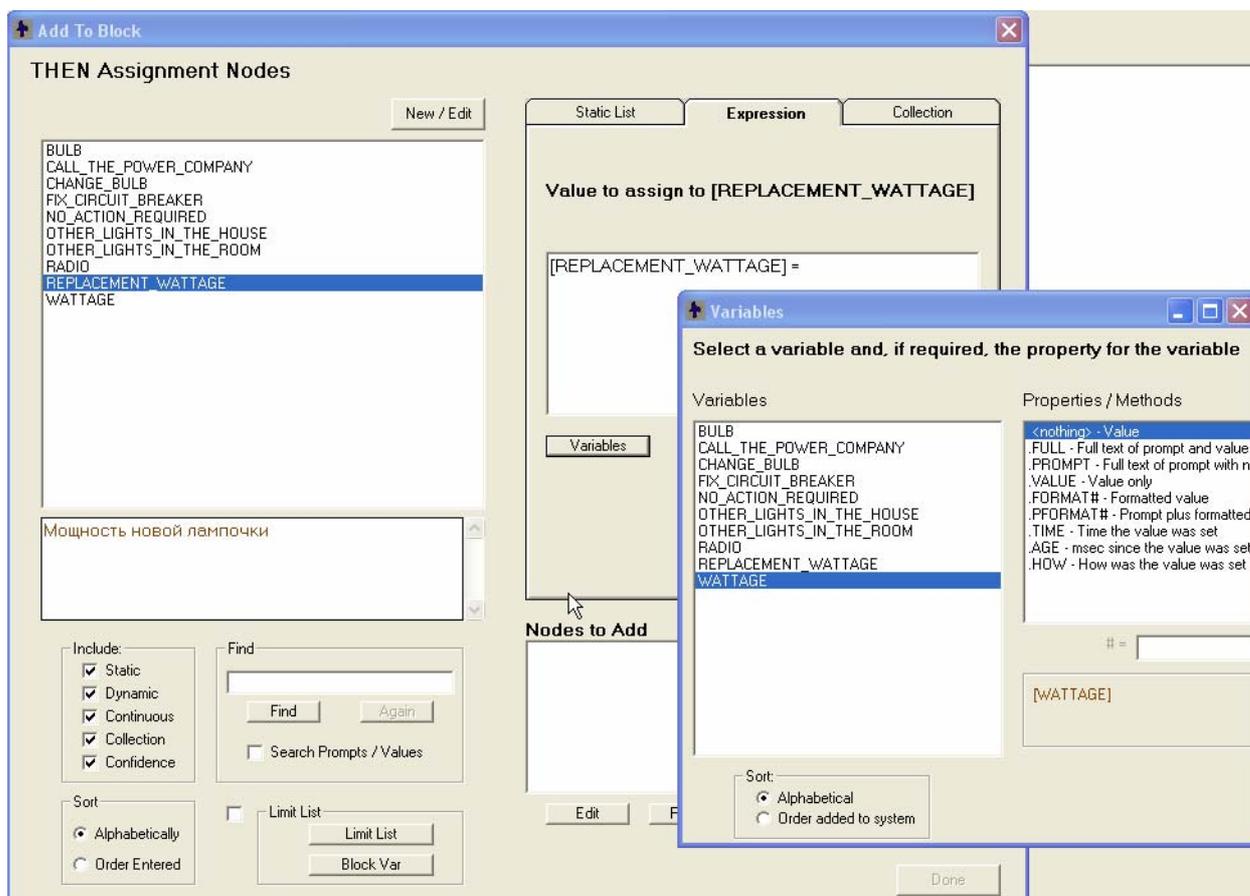
Кнопка Functions позволяет добавлять любую из математических или строчных функций которые поддерживаются в CORVID.

Примечание:

CORVID проверяет синтаксис всех новых выражений, которые добавляются в систему.

Выражение в поле редактирования правильное, но если, допустим, добавить "X" в конце выражения, что является синтаксической ошибкой, нажать кнопку Add to List.

Отобразится окно синтаксических ошибок. Оно показывает, что есть ошибка в выражении. В нем любые числовые части выражения преобразованы на "#" а строчные выражения в "\$". Это уменьшает размер и делает поиск проблем в выражении легче. В данном случае, "X" несомненно, проблема его нужно удалить и нажать кнопку Recheck. Теперь выражение добавлено в Node to Add список. Выражение может редактироваться в окне Syntax Error, которое также имеет кнопки Variables и Functions, чтобы помочь в построении выражений



Нажмите Done, чтобы добавить новый узел к Логическому Блоку. Это завершает блок, который установит величину мощности сменной лампочки.

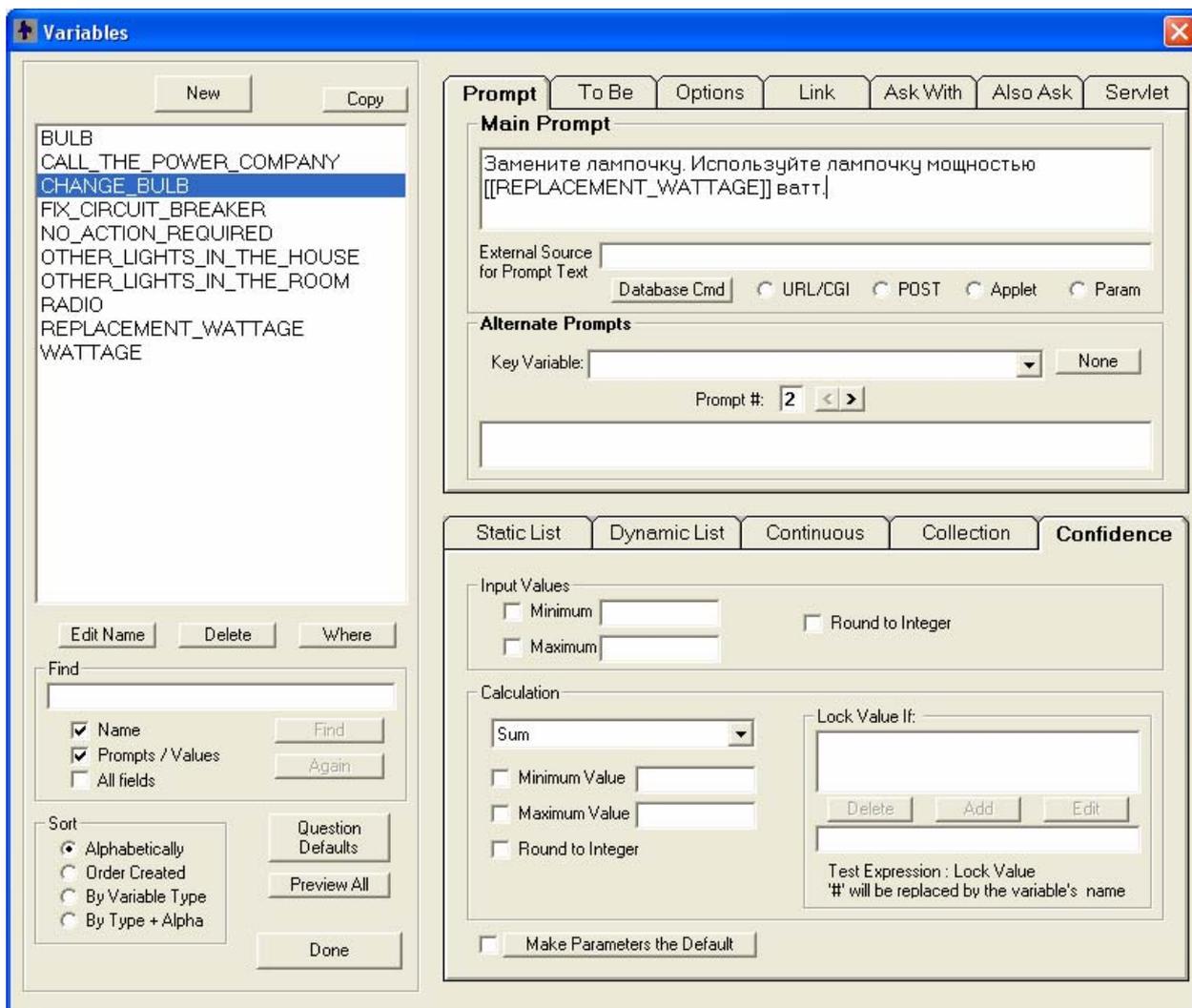
Шаг 9

Следующий шаг, отобразить эту переменную в Результатах. Это могло бы быть сделано посредством модифицирования результатов, но есть путь легче. переменные CORVID обычно используются в выражениях в квадратных скобках []. Они также могут быть вставлены в текст, будучи установленными в ДВОЙНЫХ квадратных скобках [[]]. Это справедливо для любой переменной.

У нас есть доверительная переменная [Change_bulb], которая используется, для отображения рекомендации смены лампочки. Все что мы должны сделать – расширить эту переменную, включив в неё информацию о мощности сменной лампочки.

Откройте окно переменных, щелкнув на иконке и выберите переменную Change_bulb. Затем в строку подсказки добавьте текст: "Используйте лампочку мощностью [[Replacement_wattage]] ватт". Когда эта переменная отображена в результатах, "[[Replacement_wattage]]" будет заменена величиной которая система установит для переменной [Replacement_wattage].

Использование двойной квадратной скобки в CORVID очень мощное средство. Текстовые строки могут динамически базироваться на системных величинах и рекомендациях.



Шаг 10

Теперь попробуйте запустить систему. Нажмите иконку Run. Как ожидалось, первые 2 вопроса - одинаковые. Отвечайте что лампочка вышла из строя, а другие лампочки в комнате остаются включенными. Это запустит правила, которые рекомендуют заменить лампочку.

Теперь, для того, чтобы отобразить рекомендации, системе нужна величина переменной [Replacement_wattage], чтобы включить ее в текст рекомендации.

Лабораторная работа 6: Переменные коллекции

Цель работы: Научиться работать с переменными коллекции.

Переменные коллекции являются специальным типом переменных в CORVID чьё "значение" является списком. Различные действия могут добавить пункты к списку, удалять пункты из списка, сортировать список или производить действия над пунктами в списке.

Переменные Коллекции очень гибкие и могут быть использованы для построения сообщений, динамических HTML форм или для хранения любого текста который может понадобится системе.

Этот раздел покажет простое использование переменных коллекции.

В нашей системе, есть правила, решающие какую мощность рекомендовать для сменной лампочки.

Допустим, мы составляем список покупок, в который можно включить лампочку и другие различные вещи которые должны быть куплены.

Правило, которое говорит о замене лампочки должно быть расширено, чтобы поместить новую лампочку в список покупок. Так как список покупок является списком свободного формата, в который можно добавлять пункты из многих источников, это идеальный пример для переменных коллекции.

IF

Лампочка должна быть заменена **AND** текущая мощность – выше 75 ватт

THEN

Замените 75 ваттной лампочкой **AND** Добавьте лампочку к списку покупок

IF

Лампочка должна быть заменена **AND** текущая мощность – 75в или меньше

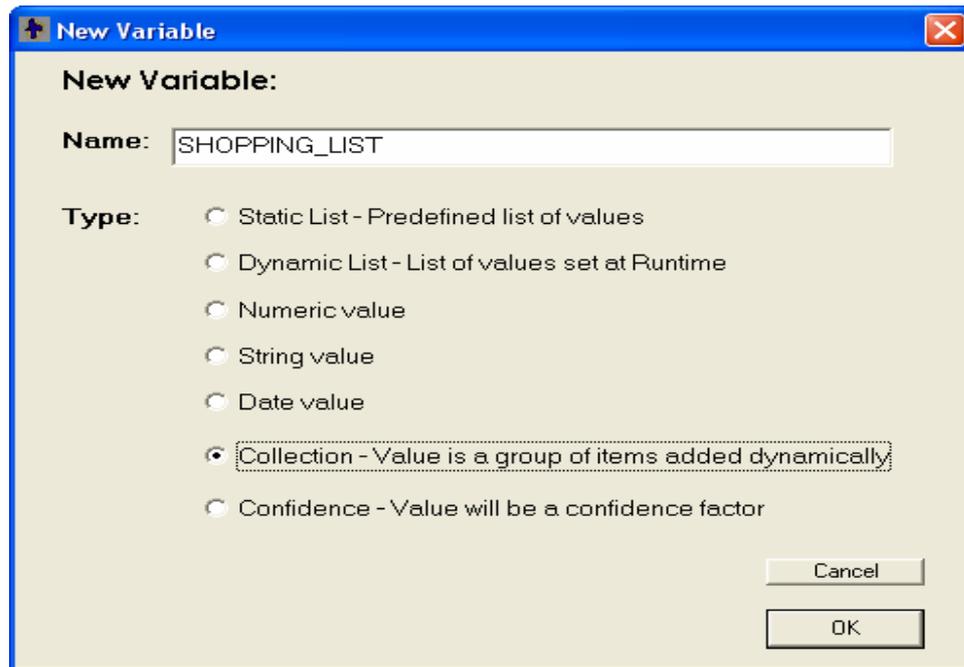
THEN

Замените лампочкой той же мощности **AND** Добавьте лампочку к списку покупок

Шаг 1

Сначала добавьте новую переменную коллекции и назовите ее "Shopping list ". В строке подсказки можно написать "Список покупок".

Переменные коллекции имеют ряд опций для инициализации их из базы данных или списка величин, но в этом случае, коллекция не должна быть инициализирована. Она будет иметь пункты, добавленные правилами в систему.

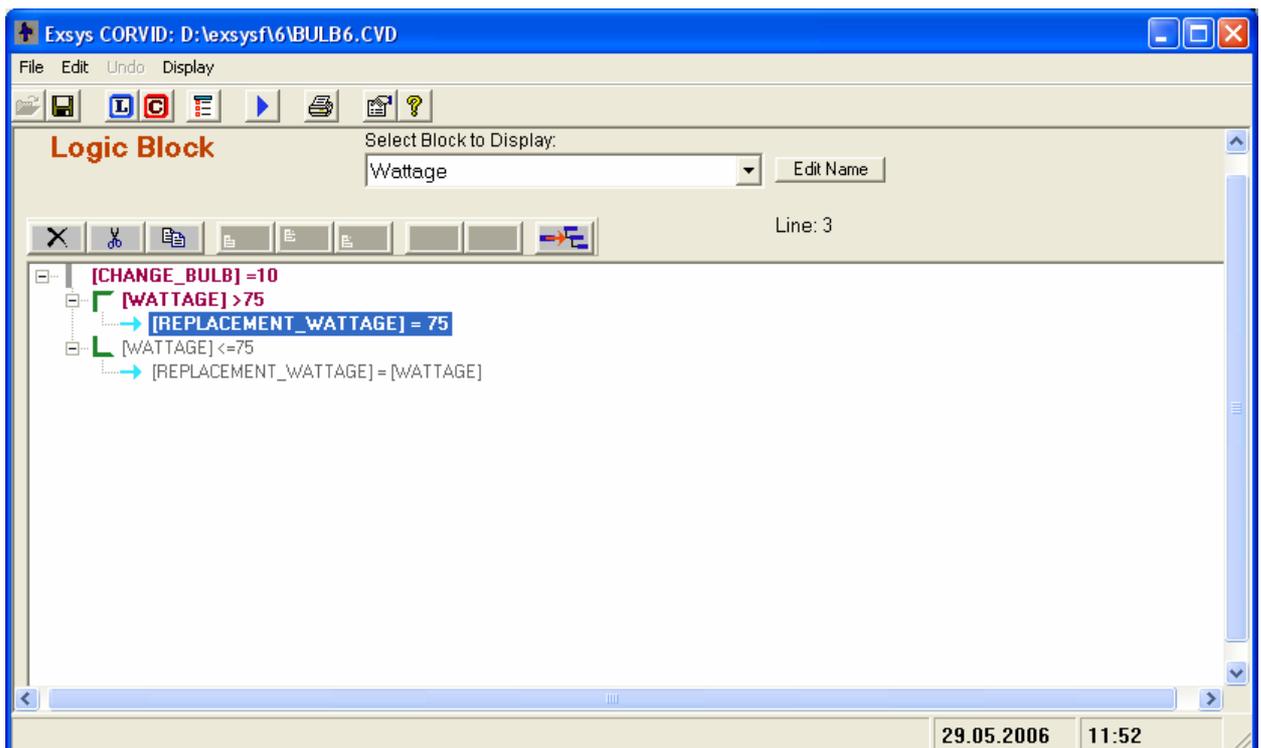


Шаг 2

Теперь, откройте Логический Блок "Wattage". Эти правила срабатывают, только если лампочка должна быть заменена.

Правила будут расширены, и в случае когда лампочка должна быть заменена, мы добавляем нужный тип лампочки в список покупок.

Щелкните на узле "[Replacement_wattage]=75", чтобы выбрать его. Затем щелкните на кнопке Then\Variable, чтобы добавить новый THEN узел.



Шаг 3

Выберете новую переменную, " Shopping_list ".

Это откроет закладку Collection, которая имеет много путей что бы добавлять или удалять пункты из переменной коллекции. "Пункты" являются текстовыми строками, и могут быть любой длины. Опции:

1. Добавлять отдельные текстовые пункты
2. Добавлять строку, которая является значением переменной
3. Сортировать пункты, по значению
4. Добавляет целый файл пунктов
5. Добавлять другую переменную коллекции
6. Удалить первый, последний или отдельный пункт, или все пункты

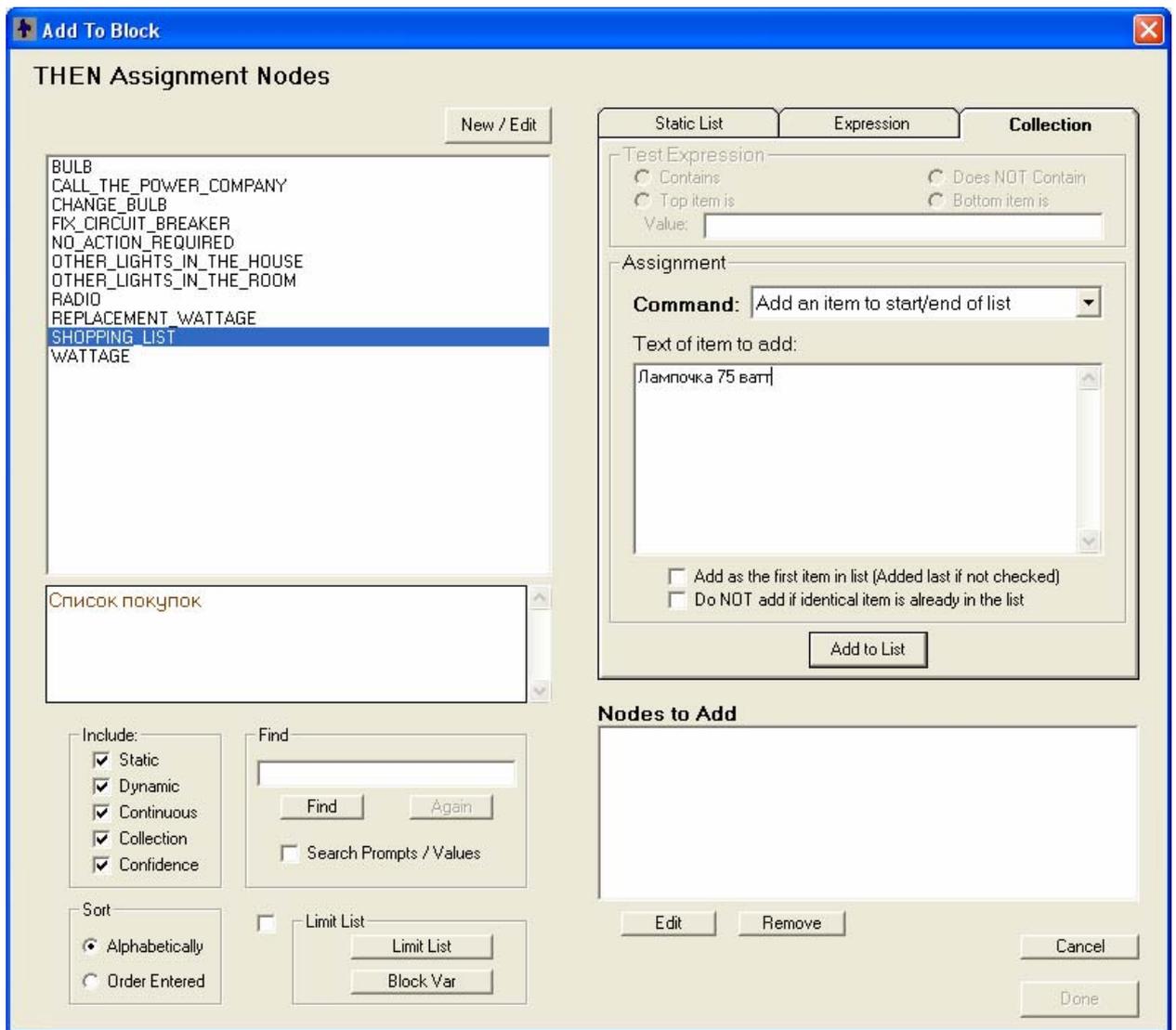
Все команды " Добавить", выбирайте только для того чтобы добавлять новые пункты в список.

В нашем случае, мы хотим добавить "лампочку 75 ватт" в список покупок, чтобы заменить ту которая сгорела.

Это просто, добавляем текстовую строку к переменной коллекции. Нам не нужно сортировать список потому что это просто список покупок.

Выберите " Add Item " из выпадающего списка и введите текст " A 75w bulb " в поле для редактирования. Щелкните кнопку Add to List.

Затем щелкните кнопку Done, чтобы добавить его к Логическому Блоку



Шаг 4

Теперь добавьте другой узел, который оперирует случаем замены лампочки менее чем 75 ватт, и добавляет ее к спискам покупок.

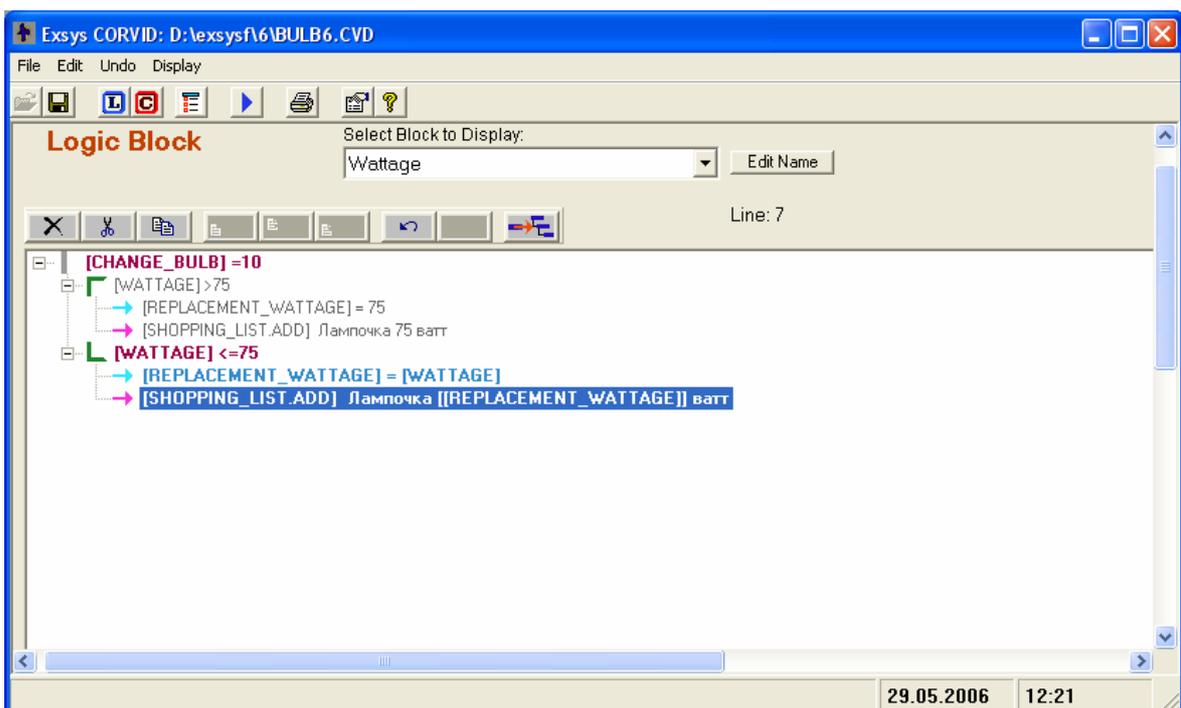
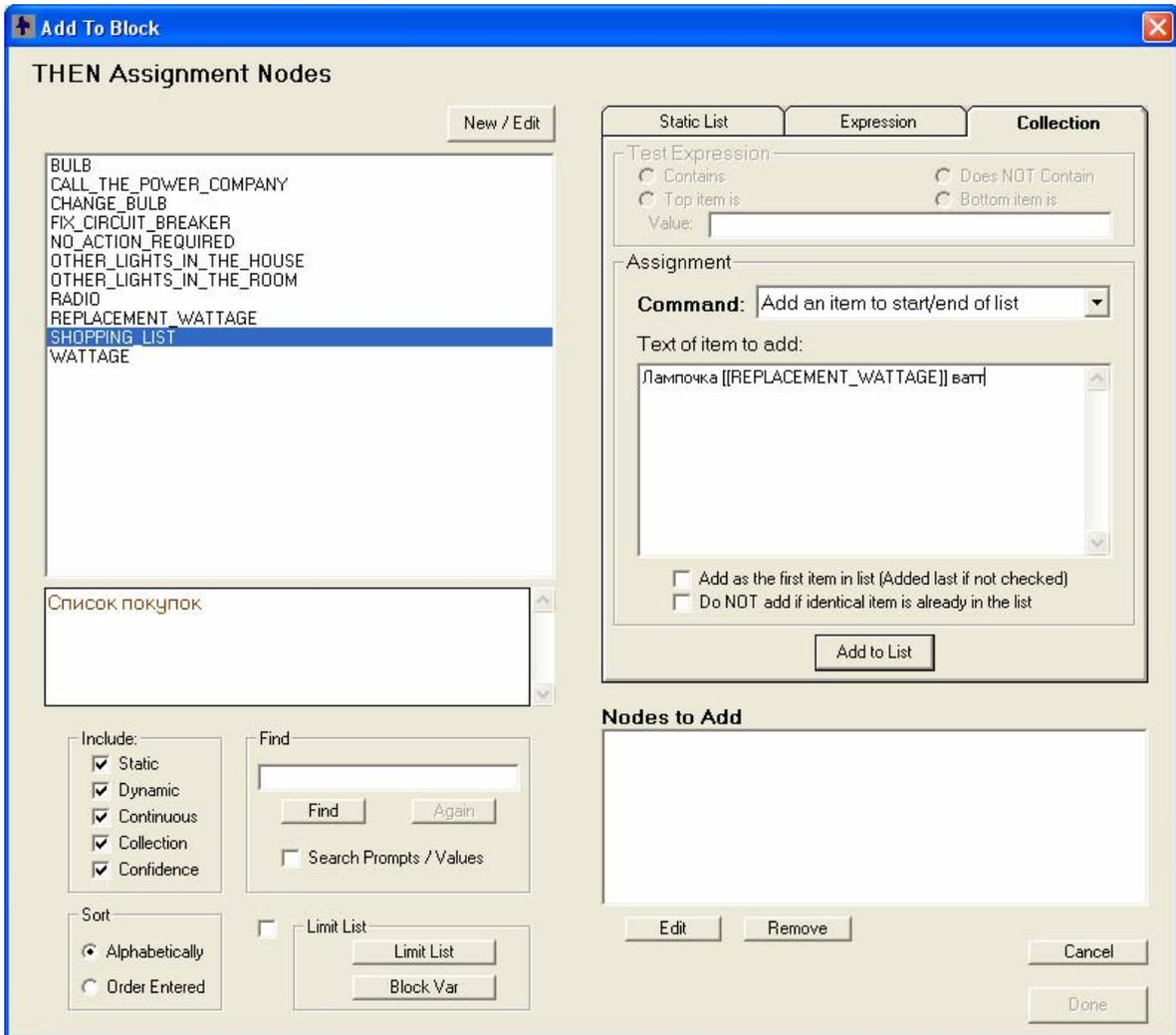
Это делается тем же путем которым добавлялся предыдущий узел. Тем не менее, на этот раз текст добавленный к переменной коллекции включает двойную квадратную скобку, чтобы вставлять значение переменной.

Использование [[]] с Коллекцией переменных очень эффективный путь строить сообщения или текстовые пояснения "свободного формата", которые содержат специфические детали пользовательского ввода или анализа.

Сначала, щелкните на узле "[Replacement_wattage]= [WATTAGE]", чтобы выбрать его. Затем нажмите кнопку Then\Variable.

На этот раз текст пункта, для добавления включает переменную в [[]]. Введите текст "Лампочка [[Replacement_wattage]] ватт" как пункт для добавления. [[Replacement_wattage]] будет заменен числовым значением этой переменной.

Щелкните Add to List кнопку и затем Done кнопку, чтобы добавить узел к Логическому Блоку.



Шаг 5

Это - все что нужно чтобы добавить пункты к списку покупок. В больших системах, должны быть другие правила, которые могли также добавлять другие типы пунктов к списку покупок. С тех пор как список является переменной коллекции, любое количество пунктов может быть добавлено в любом порядке.

Прежде, чем мы запустим систему, измените результаты в `display commands`, чтобы включить список покупок.

Добавьте пустую строку для расстояния, заголовок "Список покупок" и затем содержание переменной коллекции [`Shopping_list`]

Шаг 6

Запустите систему и посмотрите результат работы.



История кафедры ПКС

Кафедра проектирования компьютерных систем была образована в сентябре 1945 года как подразделение нового факультета электроприборостроения (позднее в 1952г. Переименованного в радиотехнический). Кафедра стала готовить инженеров, специализирующихся в новых направлениях радиоэлектронной техники, таких как радиолокация, радиоуправление, теленаведение и др. Организатором и первым заведующим кафедрой был профессор Зилитинкевич С. И., который является крупнейшим деятелем в области радиотехники и электроники, автором ряда важнейших исследований и открытий. Наиболее выдающимся из них были обнаружение «собственных колебаний электронов» в электронных лампах. На этой основе им были получены впервые электромагнитные волны дециметрового диапазона, что явилось важнейшим вкладом в возникновение и развитие современной техники сверх высоких частот. Название кафедры в тот период открыто не упоминалось, а она имела номер 11.

Профессор Зилитинкевич С. И. Руководил кафедрой с 1945 по 1951 год. Коллектив кафедры номер 11 формировался несколько лет. Первыми её преподавателями были А. Н. Иванов и А. И. Сапетин. Оба пришли в институт после демобилизации из армии, оба защищали Ленинград. А. Н. Иванов с первых дней войны был в штабе ПВО города, работал на первых отечественных РЛС. Он относился к числу тех специалистов, которые заставили поверить руководство фронта в надёжность и эффективность нового средства наблюдения за самолётами противника вместо распространённого тогда метода звукоулавливания. Первым штатным преподавателем был выпускник ЛИТМО 1945 года, прошедший специальную подготовку К. Е. Медведев - будущий доцент и декан факультета.

В течение первого года своего существования кафедра развивалась чрезвычайно быстро. Из вновь привлечённых преподавателей прежде всего следует отметить профессоров Б. А. Остроумова и Л. Б. Смияна. С 1946 года к работе на кафедре № 11 приступили старший преподаватель А. А. Тударовский, ассистенты Л. А. Гирелик, К. Г. Шаров, старшие лаборанты Г. В. Метр и П. Л. Косьмин. Основной лабораторной базой в то время были радиолокационные станции типа <Вюрсбург> (снятая с немецкого поезда) и первая отечественная станция типа <Пегматит>. Лишь в пятидесятые годы появились на кафедре действующие отечественные станции <Мист-2>, <Кобальт>, <П-8> и ряд других.

С 1951 года по 1954 кафедру возглавлял крупный специалист в области передающих устройств РЛС, один из ведущих работников радиопромышленности, кандидат технических наук, доцент А. И. Лебедев - Карманов (по совместительству). В 1954 году А. А. Тударовский, ставший к этому времени доцентом был избран заведующим кафедры № 11.

Постепенно состав кафедры начал пополняться её молодыми выпускниками. Курс «Антенны и распределение радиоволн» стал читать старший преподаватель Н. Н. Филиппов, курс «Радиолокационные системы» – выпускник кафедры 1948 года, закончивший аспирантуру в ЛЭТИ, к. т. н. А. Н. Гарпина - Домченко, курс «радиоприёмные устройства» Б. Н. Меньшов, курс «Электропитание радиоустройств» Н. В. Ефимов. Одновременно на кафедру поступило новое оборудование, в том числе современная измерительная аппаратура, что позволило создать собственную лабораторную базу по всем курсам. Большую работу провели ассистент Л. В. Шенников, старшие лаборанты Э. С. Кочанов, Г. Н. Грязин, К. Г. Шаров, ставший затем заведующим лабораторией.

На кафедре № 11 проводилась также большая научно-исследовательская работа. Так, в 1952 - 1953 годах по заказу Военно-медицинской академии был разработан и изготовлен первый отечественный электрокардиограф. В 1955 - 1957 годах под руководством заведующего кафедрой доцента А. А. Тудировского по заказу Танкового КБ проводились работы, связанные с созданием радиолокационной Техники 8 мм. диапазона.

В 1957 году кафедры № 11 и № 76 были реорганизованы. На их базе радиолокационных приборов и устройств (РЛПУ) и радиоприёмных и радиопередающих устройств (РППУ). Первую кафедру возглавил крупный специалист в области телевидения и авиационной радиолокации, заведующий лабораторией одного из ОКБ К.Н.Т. Б. С. Мишин. С приходом Б. С. Мишина стало развиваться телевизионное направление в деятельности кафедры. Следует отметить, что он занимался телевидением ещё в 30-е годы, будучи студентом Томского государственного университета и практикантом находившейся в Ленинграде центральной радиолоборатории. В конце 30-х годов он работал на заводе имени Козицкого, где был одним из создателей первого серийного телевизора Т-1 и организации его производства. В годы войны он стал заниматься радиолокационной тематикой. В это время на кафедре развивалась работа по организации учебной и научно-исследовательской лаборатории телевидения, в которой принимал участие старший преподаватель А. И. Сапетин, занимавшийся телевидением и фотографией ещё в предвоенные годы, ассистент Г. Н. Грядин и А. В. Краситкин. Позднее к ним присоединился старший лаборант В. М. Таукчи. Под руководством Б. С. Мишина, на кафедре была выполнена первая НИР по телевизионной тематике - разработали и изготовили действующий макет подводной установки СТУ-1. Второй крупной работой было создание

приёмо-передающей быстродействующей двутелеграфной аппаратуры с плоской развёрткой. Общее руководство этой НИР осуществляли профессор М. М. Русинов и доцент Б. С. Мишин. В 1965 году на кафедре был изготовлен телевизионный стробоскоп. Это привело к новому направлению в прикладном телевидении, получивших теоретическое обоснование в докторской диссертации Г. Н. Грядина. В это время состав кафедры пополнился доцентами И. В. Ивановым, И. Ю. Рагинским, В. А. Смирновым, старшим преподавателем А. А. Зелетенкиевичем и ассистентом Ю. Н. Пановым. Был принят на кафедру только что, окончивший аспирантуру при кафедре, выпускник этой же кафедры, кандидат технических наук В. С. Салтыков. Заведовали лабораторией Н. В. Александров и затем В. М. Лакунин. В качестве инженера НИСа был принят А. В. Панков, впоследствии ставший доцентом, который только что закончил эту кафедру.

После ухода Б.С.Мишина в 1964 году пенсию, кафедру в течение года возглавлял пришедший из ВНИИ телевидения доцент И.П.Захаров. После скоропостижной кончины И.П. Захарова заведующим кафедрой был назначен доцент А.Н.Иванов.

В начале 60х годов в отечественной промышленности обнаружилась острая нехватка конструкторов и технологов радиоаппаратуры особенно специалистов в области микроминиатюризации. В связи с этим кафедра была переименована в кафедру «Конструирования и производства радиоэлектронной аппаратуры» и начала подготовку инженеров по этому направлению. Были введены новые курсы. Прежние курсы были сняты или существенно сокращены или изменены. Потребовалась коренная реорганизация лабораторной базы и переезд преподавательского состава. Новые курсы пришлось готовить и организовывать лабораторную базу ведущим доцентом кафедры Г.Н.Грязнину, В.С.Салтыкову и В.А. Смирнову. Была организована научная работа по этому направлению, в частности был заключен большой договор с Петродворцовским часовым заводом на разработку электронного хронометра в микроминиатюрном исполнении. По окончании этой работы на данный хронометр были получены несколько авторских свидетельств.

В 1970 году радиотехнический факультет ЛИТМО был ликвидирован. Кафедру КиПРЭА переименовали в кафедру «Конструирования и производства электронно-вычислительной аппаратуры» (КиПВА) и перевели на факультет точной механики и вычислительной техники. Коренной переделке подвергся учебный план по которому велась подготовка специалистов. Были выделены два основных направления: автоматизация конструирования ЭВА и технология производства микроэлектронных устройств ЭВА.

В конце 1973 года на должность заведующего был избран д.т.н. профессор Ф.Г.Старос. Профессор Ф.Г.Старос являлся одним из основных родоначальников отечественной микроэлектроники. Он был главным разработчиком Советского центра микроэлектроники в Зеленограде. Под его руководством был разработан и издан первым в мировой практике прообраз персонального компьютера - УМ-1-НХ. В связи с назначением профессора Ф.Г.Староса директором одного из институтов А.Н.СССР во Владивостоке в начале 1974 года, он вынужден был покинуть кафедру КиПЭВА и на должность заведующего кафедрой был избран выпускник ЛИТМО 1959 года к.т.н. доцент В.В.Новиков (впоследствии д.т.н. профессор). С приходом В.В.Новикова резко усилилась работа в области микроэлектроники, были открыты научно-исследовательские темы по применению новых физических принципов при разработке различных электронных устройств. Большое участие в этих разработках принимали доценты А.В.Панков и В.С.Салтыков. В это время на фабрику был принят специалист в области схмотехники доцент В.Г.Фейгельс и ведущие специалисты промышленности, занимающиеся микроэлектроникой профессор Я.М.Беккер и доцент А.М.Скворцов (впоследствии профессор). Была налажена работа с промышленностью. Так, по договору с заводом «Россия» на территории кафедры был открыт промышленный участок по разработке и производству современных пленочных микросхем. Было установлено современное оборудование для вакуумного напыления, открыта линия по фотолитографии и т.д. На этом участке помимо решения чисто производственных задач, проводились лабораторные и научно-производственные работы для студентов и преподавателей кафедры, а сотрудники завода «Россия» работавшие на этом участке привлекались для работы на кафедре (руководство курсовым и дипломным проектированием, руководство лабораторными работами и др.).

С 1976 по 1996 кафедрой руководил известный специалист в области автоматизации проектирования электронных устройств профессор Г.А.Петухов (с небольшим перерывом когда с 1988 по 1992 год кафедру возглавлял ученик Г.А.Петухова профессор С.А.Арустамов, который в дальнейшем ушел из ЛИТМО , в связи с переходом на другую работу). За это время из других подразделений института доцент А.Л.Кузнецов и к.т.н. С.Ю.Яковлева. Получило дальнейшее направление развитие автоматизации проектирования. Был создан один из первых в ЛИТМО собственный машинный класс. Научная работа была в основном сконцентрирована в области САПР. Так в это время на кафедре проводилась большая научно-исследовательская работа по автоматизации топологического проектирования БИС на базовых кристаллах, которую возглавляли профессоры С.А.Арустамов и Г.А.Петухов.

В эти годы коллектив преподавателей пополнялся ее выпускниками: Н.С.Кармановским, Ю.А.Гатчиным, Б.А.Крыловым, К.О.Ткачевым,

В.А.Ткалич, Е.К.Фролковой. Пришла на кафедру выпускница ЛИАПа Н.Ю.Иванова, а в 1988 году кафедра была переименована в кафедру микроэлектроники и автоматизации проектирования (МАП).

С 1996 года кафедру возглавляет ее воспитанник доцент Ю.А.Гатчин. Помимо традиционной подготовки инженеров конструкторов-технологов по микроэлектронике и автоматизации проектирования вычислительных средств (специальность 2205), была начата подготовка специалистов по специальности 0754 «Комплексная защита объектов информатизации», причем основное внимание уделяется программно-аппаратной защите информации компьютерных систем.

В 1998 году кафедра была переименована и получила название «Кафедра проектирования компьютерных систем», что отразило содержание основных научных исследований и направления подготовки студентов и аспирантов.

За прошедшие годы кафедра подготовила более 4000 дипломированных инженеров. Более 50 молодых ученых защищали кандидатские диссертации, 10 человек защитили диссертации на соискание ученой степени доктора наук. Выпускники кафедры работают в ведущих научных центрах и учебных заведениях России, Европы, Азии и Америки в промышленных и коммерческих фирмах, лабораториях и кафедрах университета.

В настоящее время кафедра является одним из ведущих российских научных и образовательных центров, ориентированным на фундаментальные и прикладные исследования в области микроэлектроники и САПР, подготовку высококвалифицированных специалистов 21 века.

РЛПУ (1945-1966) Решением Правительства в августе 1945 года в ЛИТМО был открыт факультет электроприборостроения. Приказом по Институту от 17 сентября 1945 года на этом факультете была организована кафедра радиолокационных приборов и устройств, которая стала готовить инженеров, специализирующихся в новых направлениях радиоэлектронной техники, таких как радиолокация, радиоуправление, теленавигация и др. Организатором и первым заведующим кафедрой был д.т.н. профессор ЗИЛИТИНКЕВИЧ С.И. (до 1951 года). Выпускникам кафедры присваивалась квалификация "инженер - радиомеханик", а с 1956 года - "радиоинженер" (специальность 0705). В разные годы заведовали кафедрой доцент МИШИН Б.С, доцент ЗАХАРОВ И.П., доцент ИВАНОВА А.Н.

- КиПРЭА** Кафедра конструирования и производства радиоэлектронной аппаратуры. Каждый учебный план специальности 0705 (1966-1970) коренным образом отличался от предыдущих планов радиотехнической специальности своей четко выраженной конструкторско - технологической направленностью. Оканчивающим институт по этой специальности присваивалась квалификация "инженер конструктор - технолог РЭА". Заведовал кафедрой доцент ИВАНОВ А.Н.
- КиПЭВА** Бурное развитие электронной вычислительной техники и внедрение ее во все отрасли народного хозяйства потребовали от отечественной радиоэлектронной промышленности решения новых ответственных задач. Кафедра стала готовить инженеров по специальности 0648, Подготовка проводилась по двум направлениям: автоматизация конструирования ЭВА и технология микроэлектронных устройств ЭВА, Заведовали кафедрой д.т.н, проф. НОВИКОВ В.В. (до 1976 г.), затем проф. ПЕТУХОВ Г.А.
- Кафедра МАП** Кафедра микроэлектроники и автоматизации проектирования выпускала инженеров конструкторов - технологов по микроэлектронике и автоматизации проектирования вычислительных средств (специальность 2205). Выпускники этой кафедры имеют хорошую технологическую подготовку и успешно работают как в производстве полупроводниковых интегральных микросхем, так и при их проектировании, используя современные методы автоматизации проектирования. Инженеры специальности 2205 требуются микроэлектронной промышленности и предприятиям - разработчикам вычислительных систем. Кафедрой с 1988 по 1992 год руководил профессор АРУСТАМОВ С.А., затем снова профессор ПЕТУХОВ Г.А.
- Кафедра ПКС** Кафедра проектирования компьютерных систем выпускает инженеров по специальности "Проектирование и технология электронных средств". Область профессиональной деятельности выпускников включает в себя проектирование, конструирование и технологию электронных средств, отвечающих, целям их функционирования, требованиям надежности, дизайна и условиям эксплуатации, Кроме того, кафедра готовит специалистов по специальности 0754 "Комплексная защита объектов информации", причем основное внимание уделяется программно-аппаратной защите информации компьютерных систем, С 1996 года кафедрой заведует д.т.н., профессор ГАТЧИН Ю.А.

За время своего существования кафедра выпустила 4023 инженера, из них по специальности 0705 - 2472 чел., по специальности 0648 (2205) - 1551 чел. и по специальности 0648 220600 – 28 чел. На кафедре защищены более 50 кандидатских диссертаций и 8 докторских.