РАЗДЕЛ X. ТЕХНИКА

Балтрашевич В.Э.

Анализ комбинации блоков ввода и логического вывода ЭС

*Санкт-Петербургский Государственный электротехнический университет
(Россия, Санкт-Петербург)*

doi: 10.18411/sr-10-08-2021-24

Аннотация

Рассматривается разработка программных моделей блоков ввода знаний и логического вывода для ЭС на правилах продукций и для ЭС на списке атрибутов для автономного и сетевого режимов работы. Предлагаются различные варианты АОС.

Ключевые слова: поверхностные и глубинные знания, экспертные системы, системный анализ, имитационное моделирование.

Abstract

The development of software models of knowledge input and logical inference blocks for ES based on production rules and for ES on the list of attributes for autonomous and network operation modes is considered. Various AOC options are offered.

Keywords: surface and in-depth knowledge, expert systems, systems analysis, simulation.

Есть два основных вида экспертных систем (ЭС): ЭС, базы знаний (БЗ) которых записаны на базе правил продукции [2]; ЭС, базы знаний которых записаны на базе списка атрибутов (без правил продукции) [1].

Так как используются БЗ разных структур, то предлагается новый вариант мнемонических сокращений ЭС и их блоков, использующий обозначение их внутренней работы (БЗ) и внешнего представления исходных данных (БЗ). Так ЭС, внутренняя работа, которой базируется на списке атрибутов, и предназначенная для обработки БЗ на правилах продукций, будет обозначаться ЭС_СА_ПП. Раньше, в основном, использовались ЭС_ПП_ПП. В работе также будут проанализированы ЭС_СА_СА.

Основным достоинством ЭС являются хранящиеся в них знания. Мы рассмотрим разработку знаний об индивидуальной (автономной) и совместной (сетевой) работе блоков ввода (БВ) и логического вывода (БЛВ) разных структур.

Системный анализ ЭС требует создания моделей блоков. В качестве формализма используем язык эксперта. Т.е. фактически мы получаем второй уровень ЭС на основе соответствующих БЗ. Алгоритмы работы этих блоков записаны на языке эксперта.

Базовую инструментальную (пустую) ЭС (ИЭС) рассматриваем как «аппаратную». Отметим, что «аппаратные» блоки записаны на языке программирования, а программные блоки записаны на языке эксперта, состоящем из поверхностных знаний (ПЗ) и из глубинных знаний (ГЗ).

Рассматриваем пару блоков, чтобы показать возможности автономной и системной работы. Создаем программные модели каждого из этих блоков с использованием ПЗ и ГЗ.

Реализуем программную модель упрощенного БВ_ПП_ПП (без конструкции «параметры» и без использования коэффициентов доверия) для того, чтобы была разница между «аппаратным» и программным вариантом БВ при имитационном моделировании (ИМ). Ограниченность программного БВ_ПП_ПП используется для

наглядности испытаний (приходится задумываться о составлении структуры анализируемой системы – какие блоки включать в систему). Так из-за ограниченности программного БВ_ПП_ПП, при моделировании программного БЛВ_ПП_ПП, для его полной проверки надо использовать «аппаратный» БВ_ПП_ПП. При использовании полного транслятора БВ_СА_ПП таких проблем нет, но возникают другие проблемы (например, вопросы стыковки).

Реализуем программную модель полного БЛВ_ПП_ПП для того, чтобы проверить его работоспособность на различных эталонных БЗ, в том числе и использующих ГЗ.

Возможно, имитационное моделирование каждого блока отдельно, но лучше их работу моделировать совместно, т.к. использование правил продукции позволяет легко модифицировать порядок ИМ. Так слегка изменяя правила, мы можем запускать работу каждого блока, как в автономном режиме, так и в сетевом режиме. Т.е. загружаем объединенную БЗ работы БО_ПП_ПП и БЛВ_ПП_ПП. Для СА приведем программные модели каждого блока. БЛВ моделировать без программного БВ нельзя, но можно использовать «аппаратный» БВ.

Поверхностные знания БВ_ПП_ПП и БЛВ_ПП_ПП. Как уже отмечалось в [2], в целях компактного представления программы на одной строке представлено несколько строк языка эксперта. Символ <ВК> обозначает нажатие клавиши <Возврат каретки>.

Возможны две цели работы этой БЗ. Цель «знания» вызывает работу БВ_ПП_ПП. Цель «запрос» вызывает работу БВ_ПП_ПП, а затем работу БЛВ_ПП_ПП. Если использовать атрибут «знания1», то можно использовать «аппаратный» БВ.

Файл *tranc_Inf_Etalon.txt* (фрагмент) (ЭС_ПП_ПП)

параметры возврат <ВК>1 возврат word no

1 если знания - введены (и знания1 – введены)

и база_данных - очищена<ВК>и цель_подготовлена - да<ВК>и цель - задана

и вывод - да<ВК>и просмотр - выполнен<ВК>то запрос - да

и печать_значения_цели - нет

вопрос цель_подготовлена<ВК>Цель подготовлена?

2 если вывод - нет<ВК>то запрос - да<ВК>и печать_сообщения - нет

3 если флаг - да<ВК>и подцель_выбрана - да<ВК>и подцель - обработана

то вывод - неопределен<ВК>и возврат - флаг

4 если подцель_выбрана - нет<ВК>то вывод - да

5 если подцель - необработана<ВК>то вывод - нет

6 если правило_найденно - да<ВК>и правило - обработано<ВК>то подцель - обработана

7 если вопрос - найден<ВК>и ответ - получен

то подцель - обработана

8 если процедура_найдена - да<ВК>и процедура – обработана

то подцель - обработана

9 то подцель - необработана

10 если значение_правила - истина<ВК>и факт - записан<ВК>то правило - обработано

11 если значение_правила - ложно<ВК>и запрет_правила - да

то правило - обработано

12 если значение_правила - неопределено<ВК>и подцель_записана - да

то правило - обработано

13 то флаг – да

параметры знания1<ВК>15 аппаратный_ввод word no

Замечание: Следующие атрибуты связаны с соответствующими процедурами

ГЗ: база_данных, цель, процедура_найдена, процедура, правило_найденно,

значение_правила, факт, вопрос, ответ, печать_сообщения, печать_значения_цели, знания1, подцель_выбрана, запрет_правила, подцель_записана

41 если ввод_слова - да<ВК>и число - да<ВК>и ввод_след_слова - да

и часть_если - да<ВК>и часть_то - да<ВК>то предложение - правило

42 если слово - если<ВК>и флаг1 - да<ВК>и выражение - да<ВК>то часть_если - да

43 если слово - то<ВК>то часть_если - да

44 если атрибут - да<ВК>и предикат - да<ВК>и значение - да<ВК>и конец_строки - да

то строка - да<ВК>и условие1_обработано - да

45 если строка - да<ВК>и след_слово - то<ВК>то выражение - да

46 если строка - да<ВК>и след_слово - и<ВК>то выражение - да<ВК>и возврат - флаг1

47 если слово - то<ВК>и строка - да<ВК>то часть_то - да

48 если след_слово - то<ВК>и строка1 - да<ВК>то часть_то - да

и условие2_обработано - да

49 если атрибут1 - да<ВК>и предикат1 - да<ВК>и значение1 - да

и конец_строки1 - да<ВК>то строка1 - да

50 то флаг1 - да

51 если ввод_слова - да<ВК>и число - нет<ВК>и слово - подсказка<ВК>и атрибут - да

и конец_строки - да<ВК>и текст - да<ВК>то предложение - подсказка

и подсказка_обработана - да

52 если ввод_слова - да<ВК>и число - нет<ВК>и слово - перевод<ВК>и атрибут - да

и конец_строки - да<ВК>и текст - да<ВК>то предложение - перевод

и перевод_обработан - да

56 если предложение - правило<ВК>то язык - эксперта

53 если ввод_слова - да<ВК>и число - нет<ВК>и слово - вопрос<ВК>и атрибут - да

и конец_строки - да<ВК>и текст - да<ВК>то предложение - вопрос

и вопрос_обработан - да

57 если предложение - перевод<ВК>то язык - эксперта

58 если предложение - подсказка<ВК>то язык - эксперта

60 если предложение - вопрос<ВК>то язык - эксперта

61 если начальная_установка - да<ВК>и флаг3 - да<ВК>и синт_анализ - да

то знания - введены<ВК>и очистка - да

62 если язык - эксперта<ВК>и конец_файла - да<ВК>то синт_анализ - да

63 если язык - эксперта<ВК>и конец_файла - нет<ВК>то синт_анализ - нет

и возврат - флаг3

64 то флаг3 - да

вопрос_конец<ВК>кончат анализ?(да/нет)

Замечание: Следующие атрибуты связаны с соответствующими процедурами ГЗ с помощью конструкции «параметры» языка эксперта:

конец_файла, начальная_установка, ввод_слова, число, ввод_след_слова, слово, атрибут, предикат, значение, конец_строки, след_слово, атрибут1, предикат1, значение1, конец_строки1, условие1_обработано, условие2_обработано, текст, подсказка_обработана, очистка, перевод_обработан, вопрос_обработан

Связь ПЗ и ГЗ осуществляется с помощью конструкции «параметры» языка эксперта, которая в нотации Бэкуса–Наура имеет вид:

<параметры> ::= параметры <атрибут><конец строки><номер

процедуры>><имя процедуры><признак типа ответа><признак записи ответа><конец строки>

<признак типа ответа> ::= <word> | <integer> | <real>

<признак записи ответа> ::= <yes> | <no>

Приводить ГЗ не будем из-за их значительного размера. Но представление об используемых процедурах дается в названиях соответствующих атрибутов.

ИМ БВ_ПП_ПП. Окно «Предметной области» – интерфейс программного БВ_ПП_ПП.

Рис. 1. Вид окна «предметной области» программного БВ_ПП_ПП

Цель «знания» задается в «аппаратном» БЛВ. Транслятор проверяет только синтаксис. Кнопка «Просмотр БЗ» позволяет просмотреть внутреннее представление введенной БЗ. Семантика проверяется с помощью дополнительного «аппаратного» БЛВ_ПП_ПП (кнопка «Экспресс проверка»).

Если с атрибутами связать «вопросы» и убрать процедуры («параметры»), то можно сделать обучающую систему, т.е. систему, закрепляющую знания. Пользователь видит файл со знаниями и отвечает на вопросы о нем. Вопросы задаются про структуру предложений языка эксперта. Пользователь подтверждает структуру и запоминает ее (закрепляет свои знания).

Текст вопросов.

вопрос ввод_слова

вы выделили первое слово в предложении?(да/нет)

вопрос число<ВК>выделенное слово является числом?(да/нет)

вопрос ввод_след_слова<ВК>вы выделили очередное слово в предложении?(да/нет)

вопрос слово<ВК>введите выделенное слово

вопрос след_слово<ВК>выделите и введите очередное слово

вопрос атрибут<ВК>следующим словом является атрибут?(да/нет)

вопрос атрибут1<ВК>следующим словом является атрибут?(да/нет)
 вопрос предикат<ВК>следующим словом является предикат?(да/нет)
 вопрос предикат1<ВК>следующим словом является предикат?(да/нет)
 вопрос значение<ВК>следующим словом является значение атрибута?(да/нет)
 вопрос значение1<ВК>следующим словом является значение атрибута?(да/нет)
 вопрос конец_строки<ВК>на этом строка кончается?(да/нет)
 вопрос конец_строки1<ВК>на этом строка кончается?(да/нет)
 вопрос текст<ВК>имеется ли строка текста?(да/нет)
 вопрос конец<ВК>кончать анализ?(да/нет)

Если сделать процедуры проверяющими, то можно сделать контролирующую АОС.

Работоспособность разрабатываемых блоков проверяется с помощью ИМ. Для проверки создаваемых БЗ нужны отлаженные (эталонные) БЗ.

Выбор эталонных БЗ. При проведении системного анализа ЭС мы будем разрабатывать БЗ, описывающие работу этих блоков. Но т.к. в процессе разработки возможны ошибки, то нам нужны проверенные исходные данные, т.е. эталонные БЗ проверенные и отлаженные из любой предметной области. Эти БЗ должны удовлетворять определенным требованиям. Как уже отмечалось, мы должны провести системный анализ структур и процессов в них протекающих. Чтобы проверить структуры, нам надо задать БЗ с ПЗ, БЗ с ПЗ и с коэффициентами доверия (КД), и БЗ с ПЗ и с ГЗ. Это БЗ для ИЭС на базе ПП. Эталонную БЗ для ИЭС на базе списка атрибутов (ИЭС_СА) приводить не будем. Будем предполагать, что используемая ИЭС_СА проверена, и мы будем разрабатывать новую БЗ и ее отлаживать для АОС БЛВ_СА_ПП.

ИМ БЛВ_ПП_ПП. Окно «Предметной области» – интерфейс программного БВ.

Рис. 2. Вид окна «предметной области» программного блока логического вывода

БВ_ПП_ПП – упрощенный, а БЛВ_ПП_ПП – полный. Чтобы его проверить полностью, надо использовать «аппаратный» БВ_ПП_ПП. Проверяем на разных эталонных БЗ (с коэффициентами доверия, с ГЗ).

Если с атрибутами связать вопросы (по аналогии с БВ) и сделать наглядным процесс логического вывода, то можно сделать демонстрационный пример системы закрепляющей знания БЛВ.

Возможны различные виды АОС, которые используют:

- только вопросы (закрепляющие знания АОС),
- вопросы и проверяющие процедуры (контролирующие АОС),
- вопросы и подсказывающие процедуры (подсказывающие АОС).

Следует отметить, что в процессе работы можем использовать несколько «аппаратных» однотипных блоков. Например, для проверки программного БВ –

экспресс проверка проводится на дополнительном «аппаратном» БЛВ. Другой «аппаратный» БЛВ – для работы с основной БЗ.

Теперь рассмотрим БВ и БЛВ, созданные на СА.

Т.к. мы фактически проводим системный анализ, то нам надо рассмотреть возможность совместной работы БЗ с разными структурами (ПП или СА).

Транслятор для ЭС_СА - это упрощенный транслятор для ЭС_ПП (отсутствие правил продукции) с некоторыми добавками (конструкция «цель» и другие). БЛВ для ЭС_СА (полный вариант) тоже значительно проще (по сравнению с ПП).

Рассмотрим АОС БЛВ на СА для исходной БЗ_ПП.

В [4] приведена БЗ АОС БЛВ для обработки БЗ, записанных на базе правил продукции, записанная на базе списка атрибутов. (АОС БЛВ_СА_ПП)

Надо отметить, что в системной части «аппаратной» ИЭС были внесены некоторые изменения. Использовались два варианта процедуры GetAnswer. Один для обработки БЗ на базе списка атрибутов, другой – для правил продукции. В первом варианте для выбора следующего целевого атрибута используется конструкция

```
if (value <> 'нет') then
    BaseZnD.GetMainGoal(pItem.NextGoal_1,BaseZnD.GoalList);
if (value = 'нет') then
    BaseZnD.GetMainGoal(pItem.NextGoal_2,BaseZnD.GoalList);
```

Во втором варианте задание следующего целевого атрибута производится с помощью обработки правил продукции.

Если из используемой БЗ в данном примере убрать вопросы и подсказывающие процедуры (т.е. обрамление, связанное с созданием системы обучения), то мы получим БЗ, описывающую работу БЛВ_СА_ПП.

Файл ІАОС_9_1.txt. БЗ для БЛВ_СА_ПП (фрагмент)

```
цель ввод_знаний
параметры ввод_знаний<ВК>151 ввод_знаний word по
вопрос ввод_знаний<ВК>вы можете ввести знания
вопрос установка_цели<ВК>вы выбрали цель вывода
параметры установка_цели<ВК>3 установка_цели word по
...
next ввод_знаний установка_цели null
next установка_цели подцель_выбрана null
next подцель_выбрана правило_найдено печать_значения_цели
next поиск_вопроса ответ_процедура_найдена
next ответ_подцель_выбрана null
next правило_найдено значение_правила поиск_вопроса
next процедура_найдена процедура_обработана поиск_вопроса
next процедура_обработана подцель_выбрана null
next значение_правила правило_неопределено null
next правило_неопределено подцель_записана правило_ложно
next подцель_записана подцель_выбрана null
next правило_ложно запрет_правила правило_истинно
next запрет_правила подцель_выбрана null
next правило_истинно факт null
next печать_значения_цели null null
```

Следует подчеркнуть, что этот вариант программного БЛВ работает и с исходными БЗ, записанными на базе списка атрибутов. Т.е. Файл ІАОС_9_1.txt является и БЗ для БЛВ_СА_СА.

Теперь приведем БЗ, описывающую работу транслятора БВ_СА_ПП

Возможны разные степени детализации. Так в файле `trans_Inf_Etalon.txt` приведена БЗ для транслятора БВ_ПП_ПП, в которой степень детализации доходила до лексемы входного файла. В ниже приведенном трансляторе БВ_СА_ПП степень детализации доходит до конструкции «предложения» языка эксперта. Но зато в этой реализации получаем полный транслятор, в отличие от транслятора БВ_ПП_ПП. Это позволит проводить ИМ программного БЛВ_СА_ПП совместно с программным БВ_СА_ПП на разных исходных БЗ, не используя «аппаратный» БВ.

Файл `trans_SA_PP.txt`. БЗ для транслятора БВ_СА_ПП. (фрагмент)

цель начальная_установка

параметры начальная_установка<БК>42 начальная_установка word no

...

next начальная_установка ввод_слова null

next ввод_слова лексема_число null

next ввод_слова1 лексема_число null

next лексема_число правило лексема_вопрос

next правило ввод_слова1 null

next лексема_вопрос вопрос лексема_подсказка

next вопрос ввод_слова1 null

next лексема_подсказка подсказка лексема_перевод

next подсказка ввод_слова1 null

next лексема_перевод перевод лексема_параметры

next перевод ввод_слова1 null

next лексема_параметры параметры лексема_рекомендация

next параметры ввод_слова1 null

next лексема_рекомендация рекомендация конец_файла

next рекомендация ввод_слова1 null

next конец_файла очистка ввод_слова1

next очистка null null

Если в этой БЗ убрать атрибуты, связанные с обработкой правил, то мы получим транслятор БВ_СА_СА.

Реализация стыковки блоков ввода и вывода в варианте СА_ПП.

Запускаем ИЭС, вводим файл `trans_SA_PP.txt`, затем вводим файл `INPR1.TXT` с ботанической эталонной БЗ. После получения эталонной БЗ во внутреннем представлении, можем запустить БЛВ_СА_ПП (кнопка «ИЭС - вывод»).

Окно «предметной области» БЗ БВ для стыковки почти совпадает с окном, приведенном на рис. 1. Отличие состоит в новой кнопке «ИЭС - вывод».

При нажатии на кнопку «Просмотр БЗ», мы увидим внутреннее представление ботанической БЗ, полученное после трансляции БВ_СА_ПП. Для проверки семантики введенных знаний нажимаем на кнопку «Экспресс проверка».

Проводятся следующие действия:

`FormInfer:= TFormInfer.Create(Application);`

`FormInfer.BZ:=BaseZnPrObl1;`

`FormInfer.ShowModal;`

`FormInfer.Free;`

То есть, создается еще один объект класса «аппаратного» БЛВ_ПП_ПП. Ему передается ботаническая БЗ и проводится проверка работоспособности ботанической БЗ. Обратите внимание на то, как просто в этом случае передается БЗ, в отличие от ниже приведенного случая стыковки БВ_СА_ПП и БЛВ_СА_ПП

Для проверки работоспособности программного БЛВ_СА_ПП нажимаем кнопку «ИЭС - вывод».

Проводятся следующие действия:

`procedure TFormPredObl.Button7Click(Sender: TObject);`

```

var BZ3: TBaseZnD;
begin
  FormEdit2:=TFormEdit2.Create(Application);
  FormEdit2.ShowModal;
  BZ3:= BaseZnPrObl2;
  FormEdit2.Free;

  FormInfer:= TFormInfer.Create(Application);
  FormInfer.BZ:= BZ3;
  FormInfer.ShowModal;
  FormInfer1.Free;
  BZ3.Free;
  ShowMessage('Вывод закончился!');
  exit;
end;

```

То есть создается еще один «аппаратный» БВ_СА_ПП. Вводим файл IAOC_9_1.txt. Создается БЗ BaseZnPrObl2. Она передается вновь создаваемому «аппаратному» БЛВ_СА_ПП и начинает выполняться файл IAOC_9_1.txt. Ввод знаний в этом случае осуществлять не надо, т.к. они уже введены и находятся в BaseZnPrObl1 и **их надо передать** БЛВ_СА_ПП для использования. Это реализовано следующим образом. В модуле PREDOBL2 вводится вспомогательная переменная var BZ:TbaseZnD. В процедуре eval_proc2 для реализации каждого действия используется конструкция with BZ do вместо традиционной with BaseZnPrObl2 do .

В процедуре (составном операторе) 151 выполняем **ключевое** действие

```

151:{ввод знания2}
  begin val:='da';
    //т.к. знания уже введены, то забираем их, а не вводим.
    BZ:= BaseZnPrObl1;
  end;

```

Далее производится логический вывод для ботанической БЗ и проверяется правильность БЗ логического вывода. Правильность ботанической БЗ уже проверена.

Следует отметить, что если не делать ключевое действие, то процесс логического вывода (описанный в файле IAOC_9_1.txt) будет «дышать» (т.е. будут осуществляться переходы, описанные в конструкции next файла IAOC_9_1.txt), но процесс решения задачи классификации из ботанической БЗ выполняться не будет. Если же сделать присваивание BaseZnPrObl2:= BaseZnPrObl1, то не будут работать оба процесса.

Окно «предметной области» для программного БЛВ_СА_ПП (модуль PREDOBL2) фактически совпадает с окном на рис. 2. Ответы о значении параметров растений задаются с помощью нажатия кнопки «Ответ». Нажатие на кнопку «Стек фактов» приводит к демонстрации результатов работы ботанической БЗ.

Рассмотрим БЗ АОС БЛВ для обработки БЗ, записанных на базе списка атрибутов, записанную на базе списка атрибутов. (АОС БЛВ_СА_СА)

В качестве исходной эталонной БЗ_СА для дихотомии возьмем БЗ из [1].

Отметим, что в системной части «аппаратной» ИЭС были внесены некоторые изменения. Использовались два варианта системной процедуры GetAnswer. Оба для обработки БЗ на базе списка атрибутов и для выбора следующего целевого атрибута используется выше упоминаемая конструкция

```

if (value <> 'nem') then
  BaseZnD.GetMainGoal(pItem.NextGoal_1,BaseZnD.GoalList);
if (value = 'nem') then

```


BaseZnD.GetMainGoal(pItem.NextGoal_2,BaseZnD.GoalList);

Они отличаются использованием разных вариантов процедуры eval_proc
procedure eval_proc(number:integer; par:domen; var par1:domen;

var val:string40);stdcall; - для реализации подсказки ответов.

procedure eval_proc3(number:integer; par:domen;

var val:string40;var KD:integer); - только для проверки ответов.

IAOC_12.txt: (фрагмент)

цель ввод_знаний

вопрос ввод_знаний<BK>вы можете ввести знания

вопрос установка_цели<BK>вы выбрали цель вывода

вопрос значение_подцели<BK>Укажите значение подцели?

...

next ввод_знаний установка_цели null

next установка_цели подцель_выбрана null

next процедура_найдена процедура_обработана поиск_вопроса

next процедура_обработана подцель_выбрана null

next подцель_выбрана значение_подцели печать_значения_цели

next подцель_записана подцель_выбрана null

next факт подцель_выбрана null

next поиск_вопроса ответ процедура_найдена

next ответ подцель_выбрана null

next печать_значения_цели null null

next значение_подцели поиск_вопроса null

Т.е. небольшая коррекция ПЗ (за счет удаления обработки правил из файла IAOC_9_1.txt) и мы получаем БЗ, описывающую обработку БЗ другого типа. Контроль знаний процесса логического вывода проводится при определении значения подцели.

При ИМ проводится взаимодействие двух процессов. Процесса логического вывода под управлением БЗ, находящейся в «аппаратном» блоке логического вывода, и процесса решения задачи дихотомии, которая использует окно «предметной области» - т.е. интерфейса программного БЛВ и окно с демонстрацией задачи дихотомии. Кнопка «Просмотр» аппаратного БЛВ_СА_СА позволяет просмотреть БЗ логического вывода в представлении структур данных. Кнопка «Просмотр» окна «предметной области» позволяет просмотреть БЗ решения задачи дихотомии в представлении структур данных. БЗ дихотомии используется в качестве исходных данных при ИМ БЛВ_СА_СА.

Рис. 3. Вывод и дихотомия записаны на СА. Диалог для вывода с возможностью подсказки правильного ответа

Рис.4. Диалог для дихотомии без возможности подсказки правильного ответа, но с проверкой правильности ответа

В данном варианте диалоги разные. Первым показан диалог, допускающий подсказку правильного ответа, вторым показан диалог, получающий ответ пользователя и проверяющий его правильность. Можно сделать их одинаковыми для обоих процессов. Т.е. имеем еще один вариант варьирования структур блоков.

Вариант использования СА в АОС методом дихотомии с диалогом, допускающим подсказку правильного ответа, рассмотрен в [1].

Возможно построение структур ЭС, использующих различные сочетания БВ и БЛВ как в «аппаратном», так и в программном исполнении, а также с использованием правил продукции и с использованием списка атрибутов. Это открывает большие возможности для имитационного моделирования и тренировок в программировании.

Выводы. В результате проведенного анализа:

- сформулированы знания (на языке эксперта) о работе БВ и БЛВ в виде ПП и СА для автономного и сетевого варианта,
- рассмотрены несколько вариантов АОС,
- реализованы варианты стыковки БЗ,
- проведено имитационное моделирование разработанных программных блоков в автономном и сетевом режиме.

Ряд проведенных разработок подтвердил и эффективно продемонстрировал полезность системного анализа для исследования многоуровневых ИЭС. Были разработаны программные модели (БЗ) отдельных блоков ИЭС, с помощью которых удалось досконально изучить их работу и ввести улучшающие характеристики в описание блоков и в их программную и «аппаратную» реализацию. В качестве формализма при создании моделей использовался язык эксперта. Для хорошо изученных задач в результате системного анализа были предложены АОС на базе списка атрибутов.

Системный анализ структур позволил:

- убрать правила продукций, но сохранить объяснения,
- предложить новую форму диалога с подсказкой ответа для АОС на базе списка атрибутов (АОС_СА),
- модифицировать систему объяснений (след вывода),
- провести анализ систем, использующих БЗ с разными структурами.

Системный анализ процессов позволил провести ИМ отдельных блоков и всей ИЭС в целом.

Использование глубинных знаний и различных режимов работы ЭС позволило реализовать разработку АОС, позволяющих на базе накопленных знаний организовать обучение начинающих аналитиков. Особое значение в этом случае приобретает наличие блока объяснений ЭС. Опыт использования АОС показал целесообразность их использования в различных предметных областях. Особое значение применение АОС находит в учебном процессе.

С научной точки зрения, открывается широкая перспектива экспериментов с разработкой различных структур и их имитационного моделирования. С практической точки зрения разработанные БЗ могут использоваться для разработки новых ЭС и АОС, решающих разные задачи из разных предметных областей.

1. Балтрашевич В.Э. Интеллектуальная АОС на базе списка атрибутов / В.Э. Балтрашевич.– Beau Bassin: LAP Lambert Academic Publishing, 2019. – 132 с. ISBN 978-620-0-00382-9 (переведена на 8 европейских языков).
2. Балтрашевич В.Э. Реализация инструментальной экспертной системы / В.Э.Балтрашевич. – СПб: Политехника, 1993.
3. Перегудов Ф.И. Введение в системный анализ / Ф.И.Перегудов, Ф.П. Тарасенко. – М.: Высшая школа, 1989.
4. Балтрашевич В.Э. Разработка АОС программного блока вывода. Наука России: Цели и задачи. Сборник научных трудов по материалам XXVI международной научной конференции 10 апреля 2021 г. Часть 2. Изд. НИЦ «Л-Журнал». 2021 с.9 - 16.