OSTIS-2012
(Open Semantic Technologies for Intelligent Systems)

УДК 004.822, 004.891

РАЗРАБОТКА ЭКСПЕРТНЫХ СИСТЕМ В СРЕДЕ MULTI STUDIO

Катаев В.А.

ОАО «Пермская научно-производственная приборостроительная компания», г. Пермь, Российская Федерация

Bravo55555@yandex.ru

В работе содержится краткое описание разработки статических экспертных и информационных систем в программной среде Multi Studio российской разработки. Среда базируется на универсальном языке Multi и универсальных Multi-сетях, являющихся комбинацией синтаксических и семантических сетей. Представлены результаты сравнения Multi Studio с известной средой CLIPS.
Ключевые слова: семантические сети, экспертные системы, CLIPS, Multi Studio.

Введение

Экспертные системы (ЭС) относятся к классу интеллектуальных систем (ИС). Проектирование ИС в настоящее время имеет несколько принципиальных проблем. В статье [Голенков, 2011] обозначены следующие недостатки современных технологий проектирования ИС:

- Технологии искусственного интеллекта (ИИ) не ориентированы на широкий круг разработчиков ИС.
- Велики сроки разработки и трудоемкость сопровождения ИС.
- Высока степень зависимости технологии ИИ от платформ их реализации.
- Низкий уровень интеграции научных и практических результатов в области ИИ.
- Современные компьютеры плохо приспособлены для решения задач ИИ.

В этой же статье предлагаются пути устранения указанных недостатков:

- Ориентация на семантическое представление знаний.
- Унификация моделей ИС.
- Блочное проектирование ИС с использованием типовых компонентов.
- Поэтапное эволюционное проектирование на основе прототипов.
- Использование одинаковых принципов при проектировании ИС и инструментальных средств их разработки.
- Разработка подсистем помощи для разработчиков ИС и для их пользователей.
- Включение в ИС подсистемы автооптимизации.
- Обеспечение максимальной независимости эволюции баз знаний (БЗ) от эволюции решателей задач.
- Разработка языка унифицированного кодирования семантических сетей.
- Добавление в машину обработки знаний (МОЗ) новых агентов, не требующих внесения изменений в действующие.
- Трактовка пользовательского интерфейса ИС как специализированной ИС, имеющей свою БЗ и свою МОЗ.

При решении указанных проблем мы исходим из того, что главным направлением развития ИС является совершенствование языка проектирования ИС и разработки эффективной среды реализации этого языка. Необходимые свойства языка:

- Высокий уровень (принципиально приближенный к естественному языку).
- Простота.
- Универсальность.
- Блочность.
- Итеративность.
- Рекурсивность.
- Синонимичность.
- Омонимичность.
- Контекстность.
- Возможность манипулирования объектами сетевых структур, как наиболее универсальных.

Необходимые свойства среды разработки ИС:

- Простота.
- Универсальность.
- Мультиплатформенность.

- Среда должна позволять эффективно обрабатывать сетевые информационные структуры (базы знаний).
- Наличие интерфейсов с основными информационными системами.

Нами предпринята попытка реализации указанных свойств в среде Multi Studio, которая базируется на языке Multi и универсальной информационной структуре, называемой Multi-сетью.

1. Среда Multi Studio

Multi Studio (MS) – это универсальный многооконный диалоговый интерпретатор команд языка Multi, работающий под управлением грамматики, написанной на самом языке Multi.

Меню и системные сообщения (например, об ошибках) – на базовом языке, которым является русский. Имеется возможность использования в качестве базового языка других естественных языков. Система позволяет обработку текстов на русском, английском и на других языках.

Multi – это высокоуровневый командный (алгоритмический) язык с функциональной парадигмой и префиксной формой записи. Его принципиальный (базовый) синтаксис:

- Multi-текст – это непустой список предложений.
- Предложение – это непустой список выражений, разделенных пробелами. В конце предложения стоит точка с запятой.
- Выражение – это объект и необязательный аргум.
- Объект может быть командой, числом, литеральной строкой, именем, датой и т.п.
- Аргум – это список выражений, заключенный в круглые скобки.
- В конце предложения закрывающие круглые скобки аргумов могут отсутствовать.

Пример:

P# ("сумма =" +(X 123 ; X (5000); (1)

где **P#** – вывод на экран объектов аргума через пробел.

Результат: сумма = 5123

Универсальность (функциональная, технологическая, предметная) системы определяется универсальностью командного языка.

Функциональная универсальность предполагает наличие в языке средств (команд, функций) для обработки данных различной логической структуры (включая базы данных, базы знаний регулярной и произвольной структуры, тексты искусственных и естественных языков). Данные и программы их обработки в системе имеют одинаковый унифицированный формат и хранятся совместно в однородной электронной памяти, в так называемом бункере.

Технологическая универсальность предполагает возможность использования одного языка на всех этапах обработки информации (включая ввод, контроль, сохранение данных, выборку их из хранилища, передачу данных и вывод пользователю).

MS можно применять для решения различных задач: при обучении информатике школьников и студентов, для вычислений, для разработки информационных и экспертных систем.

Сеанс работы с MS называется беседой и состоит из серии заданий (Z). Задание на языке Multi записывается в окно ввода. Результат выполнения Z, как правило, выдается в окно вывода.

В системе также имеется возможность включать "ассистентов", которые синтезируют речь на естественном языке и озвучивают работу MS.

2. Multi-сеть представления знаний в Multi Studio

На логическом уровне Multi-сеть (МС) является синтаксической сетью (СинС), на которую накладываются семантические сети (СемС). СинС состоит из синтаксических узлов и синтаксических связей (ребер) между ними в соответствии с базовым синтаксисом языка. В каждом логическом узле находится один логический объект. Синтаксические связи (отношения) между узлами (объектами) являются бинарными направленными и в исходном тексте представлены разделителями:

Открывающая скобка – "родитель→ребенок"

Пробел между объектами – "брат→брат"

Закрывающая скобка – "брат→пусто"

Часть СинС, состоящая из узла-имени и его потомков всех уровней, называется "семья" (S). Если у имени нет потомков, то S называется одиночной. В составе S могут находиться другие S. Если в числе потомков содержится имя, совпадающее с именем главы семьи, такая S называется рекурсивной. У нескольких S могут быть общие потомки.

Пример. В задании (1) X(5000) является S.

На физическом уровне основной вид памяти системы – Бункер, файл, состоящий из ячеек ("нейро"). СинС представлена в Бункере узлами-объектами, между которыми имеются ссылки. Каждый узел занимает одно нейро.

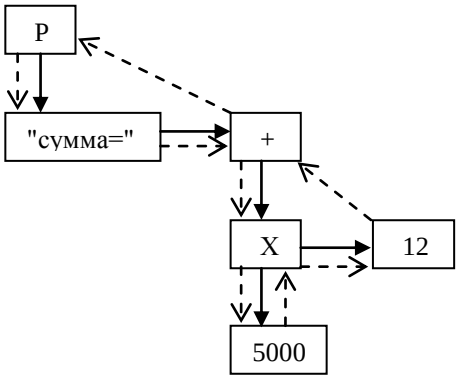

Рис.1 СинС задания (1)

Пример. Задание (1) в Бункере представляется в виде следующей СинС (рис.1).

МОЗ производит выполнение Z, используя ссылки (→) и стек. Последовательность обхода узлов Z (по принципу "сверху вниз и слева направо") указан пунктирными линиями.

Бункер имеет множество каталогов, каждый из которых представляет независимую виртуальную область Бункера – "сферу" (КС), которую также можно назвать семейным архивом, так как в каталоге сохраняются S. В каждый момент времени одна из сфер является текущей (ТС). При выполнении задания между Z и КС производится семейный S-обмен:

- Если одна сторона имеет одиночную S, а другая одноименную полную S, то потомки становятся общими.
- Если в КС нет одноименной S, то эта S помещается в КС.
- При наличии у обеих сторон одноименных полных S замена или не замена потомков S решается в зависимости от контекста.

По времени сохранения семей в каталогах сферы делятся на 3 типа:

- Каталог очищается после выполнения каждого задания.
- Каталог очищается в конце беседы.
- Каталог очищается по программе пользователя.

Семейные имена подразделяются на 2 типа: глобальные и локальные.

Соответственно S подразделяются на глобальные и локальные. Область действия глобальных имен – сфера, локальных – задание. Возможно вхождение одних S в состав других S в произвольной комбинации. Указанные свойства S позволяют использовать в сетях омонимы.

Переход из одной сферы в другую производится программно. При создании нового Бункера каталоги сфер пустые.

У каждой сферы своя СинС, на верхнем уровне которой находятся семейные имена. На одну СинС может "накладываться" множество независимых СемС. В отличие от СинС структура СемС может быть весьма разнообразной. Даже для одной СемС в общем случае возможно множество вариантов. Так в примере (1) возможна следующая эквивалентная СемС:

P# ("сумма =" +(123 X (5000) ; (2)

Здесь P# является синтаксическим родителем команды '+'. На семантическом уровне P# является функцией, а '+' – ее аргументом. Аналогично синтаксически X является отцом числа 5000, а семантически он является подстановочным идентификатором для числа 5000.

В общем случае семантические связи между объектами представляются объектами-связями. Например, предложение «Николай является отцом Маши и мужем Зои» на Multi может выглядеть так:

Николай(являться(отец(Маша) муж(Зоя; (3)

Еще пример:

Нож(состоять_из (рукоять лезвие; (4)

В частности, семантические связи между объектами в Бункере могут быть наложены по умолчанию на синтаксические связи без наличия объектов-связок.

Дети (Иван(2000 169) Маша(1998 162); (5)

В соответствии с внешней семантикой пользователя 2000 – год рождения, 169 – рост в сантиметрах.

Язык также позволяет записать это предложение с явной пользовательской семантикой:

Дети(Иван(2000:год_рождения 169:рост_см) (6)
Маша(1998:год_рождения 162:рост_см);

При этом можно писать параметры детей в произвольной последовательности и даже в неполном составе:

Дети(Иван(2000:год_рождения) (7)
Маша(162:рост_см 1998:год_рождения;

Исходные тексты программ оформляются на Multi аналогично описанию БЗ и точно также записываются в Бункер.

Таким образом, в Multi Studio моделью представления знаний является универсальная Multi-сеть.

В составе Multi-сети могут находиться фреймы и продукционные правила.

Используя команду ## [структурный вывод] пользователь может распечатать структурно (по уровням) всю сеть объектов заданной сферы или любой ее части (если он имеет доступ к ним). При этом информация выводится в синтаксисе Multi, что позволяет читать программы и данные (и обмениваться ими) при отсутствии исходных текстов.

Наличие в КС S позволяет вызывать из Бункера на исполнение поименованные программы или их части.

Так если поименовать первое предложение примера (1) как ZZ и выполнить задание

ZZ(P# ("сумма =" +(X 123 ; (8)

то в следующем задании можно запустить программу-семью ZZ:

{} (ZZ; (9)

Можно также запустить ZZ с заменой X:

{}(ZZ; X(20.55; (10)

В системе MS применяется словарь команд.

Пользователь может менять некоторые параметры словаря: заменять обозначения команд или вводить дополнительные обозначения – синонимы. Например, команде '+' можно назначить русский синоним 'сумма' или английский 'add'. Таким образом, в Multi-сети могут находиться

семантические синонимы, синтаксически обозначаемые различно.

Большинство команд Multi имеют глубокую контекстность (область действия) и вкладываюся друг в друга, что существенно упрощает, сокращает Multi-текст и увеличивает быстродействие системы.

3. Разработка экспертных систем в среде Multi Studio

Имеющийся набор команд языка Multi позволяет весьма просто разрабатывать в среде Multi Studio статические экспертные системы (ЭС). Для этого помимо общих команд (например, арифметических) имеется несколько специальных команд.

Основная команда называется «меню» и имеет следующую структуру:

menu.(заголовок" вариант1 (аргум1) вариант2 (аргум2) …) (11)

Команда меню открывает окно меню (рис.2), в котором указывается заголовок, поля вариантов и 1 или 2 дополнительных поля. Заголовок может быть наименованием проблемы или вопросом.

Поле варианта может быть разделом проблемы (ответом на вопрос в заголовке) или наименованием справки. Наименование справки – это литерал, первым символом которого является знак '!'. Поле других вариантов можно задавать именем, числом, текстом, символами + - = >= < <= != . Символы + - могут интерпретироваться как «ДА» и «НЕТ».

При выборе варианта происходит переход на аргум соответствующего варианта. В частности, аргум может содержать переход на блок команд, в котором содержится новое меню.

Если выбрано поле справки, то после окончания работы аргума происходит автоматический возврат на текущее меню.

Дополнительные поля: «назад» и «конец». Поле «назад» открывает предыдущее меню (если оно имеется). Поле «конец» открывает дополнительное меню, в котором имеется несколько вариантов возврата, вплоть до выхода из экспертной системы.

Пример фрагмента ЭС поиска неисправностей двигателя:

```
P#( «ЭКСПЕРТНАЯ СИСТЕМА
ПОИСКА НЕИСПРАВНОСТЕЙ ДВИГАТЕЛЯ» _
Состояние( menu.("Состояние двигателя?"   (12)
"НОРМА" («хорошо» Решение1)
"НЕ ЗАВОДИТСЯ" ( Вал )
«ПЕРЕБОИ» ( Мощность ) ;
Решение1(menu.rgb("255.." "255.."
munu.(«РЕШЕНИЕ 1 :»
«РЕМОНТ НЕ НУЖЕН»;
Мощность( menu. ( "Мощность двигателя?"
"ВЫСОКАЯ" ( инерция )
"НИЗКАЯ" ( контакты ) ;
```

При запуске этого задания на экране системы появляется диалоговое окно следующего вида (поля «назад» нет, так как меню первого уровня.)

При выборе варианта Пользователь может "кликнуть" на нужное поле выбора или нажать на клавиатуре цифру – порядковый номер поля. Номера имеют только первые 9 полей (1-9). После выбора ответа команда запускает соответствующее действие.

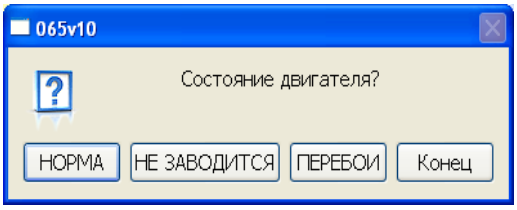

Рисунок 2. Пример меню первого уровня

При выборе варианта 1 (НОРМА), открывается новое меню «РЕШЕНИЕ 1 :», в котором тексты вопроса и первого (и единственного) ответа – красные. После выбора ответа Пользователем, система завершает работу с ЭС.

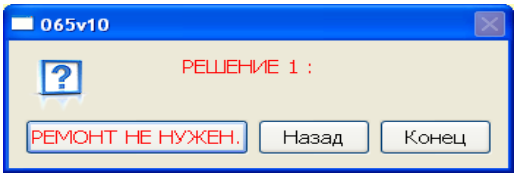

Рисунок 3. Пример цветного меню

При выборе варианта 3 (ПЕРЕБОИ) появляется окно с меню «Мощность двигателя?» и так далее до получения очередного решения (рекомендации) системы.

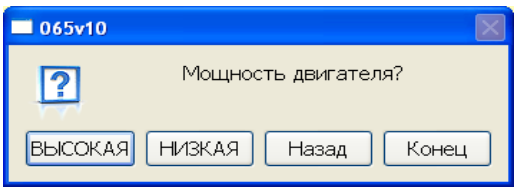

Рисунок 4. Пример меню второго уровня

Если несколько вариантов требуют одни и те же действия, их можно объединить в один составной текст-литерал, в котором тексты вариантов разделяются вертикальной чертой '|'.

```
menu. ("цвет яблока?"                          (13)
      "красный|желтый"( "ешь")
      "серый|с пятнами"("выкинь");
```

Пример справки: (14)

```
menu. ( «Куда пойдешь?» «Налево» «Направо»
«! Справка»( «Налево пойдешь – коня потеряешь.
             Направо пойдешь – Змея-Горыныча
             повстречаешь.» ) ;
```

Максимально возможное количество вариантов в одном меню – 40. В общем случае они располагаются в виде матрицы G (горизонталь) x V (вертикаль). По умолчанию в каждой строке по горизонтали размещается 6 вариантов. Дополнительная команда menu.g позволяет изменять размер G от 1 до 10.

К дополнительным командам относятся теги – это группа команд редактирования текста заголовка, вариантов (а также текста протокола работы ЭС). Например, окраска текста заголовка и

вариантов регулируется командой-тегом menu.rgb (“К.З.С”), где К указывает насыщенность красного (0-255) в общем цвете, З – зеленого и С – синего. Окраска текста протокола регулируется аналогичной командой rgb.

Теги могут располагаться в тексте задания:

- до команды «меню»
- в аргуме меню
- или отсутствовать.

В последнем случае цвета, размеры и другие свойства текстов назначаются по умолчанию.

Команды menu.sv и menu.nv позволяют запоминать текст и/или номер варианта с заданными именами, чтобы использовать их далее в текущем или следующих заданиях.

В процессе работы с окнами меню может формироваться протокол, текст которого в режиме реального времени выводится в окно вывода. Пользователь имеет возможность следить за ходом логического вывода решения и своевременно исправлять допущенные ошибки путем возврата на предыдущие меню. В конце работы протокол может выводиться на печать. Имеется 3 режима протоколирования (полное, частичное и блокировка).

Пример протокола запуска фрагмента ЭС (12):

«ЭКСПЕРТНАЯ СИСТЕМА
ПОИСКА НЕИСПРАВНОСТЕЙ ДВИГАТЕЛЯ
Состояние двигателя? ПЕРЕБОИ
Мощность двигателя? Назад
ВОЗВРАТ на предыдущий вопрос
Состояние двигателя? НОРМА
хорошо
РЕШЕНИЕ 1 : РЕМОНТ НЕ НУЖЕН.»

ЭС в Multi Studio – это сеть команд menu. и их составляющих. В приведенных примерах формируется ЭС с прямым логическим выводом. Команда меню позволяет также формировать ЭС с обратным логическим выводом.

4. Сравнение системы Multi Studio и среды CLIPS

Для сравнения Multi Studio с другими средами была выбрана система CLIPS по причине ее доступности. К тому же эта среда является базовой в российском вузовском образовании по ЭС [Сидоркина, 2011, Частиков и др., 2003].

Сравнение систем проводилось на примере тестовой ЭС поиска неисправностей двигателя, заимствованного у разработчиков системы CLIPS [Джарратано, 2007].

Пример: фрагменты экспертной системы поиска неисправностей на языке CLIPS:

```
(deffunction ask-question (?question $?allowed-values)
        (printout t ?question)
        (bind ?answer (read))
        (if (lexemep ?answer)
        then (bind ?answer (lowcase  ?answer)))
(while (not (member ?answer ?allowed-values))
(printout t ?question)
 (bind ?answer (read))
(if (lexemep ?answer)
then  (bind  ?answer (lowcase ?answer))))
 ?answer )
 (defrule determine-engine-state ""
        (not (working-state engine ?))
        (not (repair ?))
        =>
(if (yes-or-no-p "Does the engine start (yes/no)? ")
          then
(if (yes-or-no-p "Does the engine run normally
(yes/no)? ")
        then
       (assert (working-state engine normal))
        else
        (assert (working-state engine unsatisfactory)))
          else
(assert (working-state engine does-not-start)))
        )
```

Предварительные результаты сравнения систем в части разработки статических ЭС:

- Система Multi Studio – универсальна (на ней кроме разработки ЭС возможно проводить другие вычислительные работы). CLIPS – специализированная оболочка ЭС.
- Диалог в Multi оконный с параллельным протоколированием каждого шага логического вывода (по усмотрению Пользователя). В CLIPS совмещено то и другое в формате командной строки.
- В CLIPS-программе нет возвратов к предыдущим меню.
- Языком интерфейса является русский (возможен английский и другие), на CLIPS – кириллица не поддерживается.
- В Multi Studio можно модифицировать программу «на ходу», заменяя отдельные именованные части, не перекомпилируя всю программу.
- Скорость работы Multi-программы значительно выше за счет того, что программа ходит по готовым путям семантической сети. CLIPS-программа по классическому алгоритму многократно обходит базу знаний в последовательном поиске нужных правил.
- Текст программы на Multi в 4 раза короче.
- в Multi-программе используются пять операторов, вместо тридцати в системе CLIPS.
- Трудоемкость разработки статической ЭС в среде CLIPS на порядок выше, чем в Multi Studio.
- Multi-программы после короткого обучения может разрабатывать эксперт, не прибегая к услугам программиста. CLIPS -программы может писать только программист.

ЗАКЛЮЧЕНИЕ

В докладе предложен вариант решения языковой проблемы разработки статических экспертных систем с использованием командного языка Multi в среде Multi Studio. Система Multi Studio развивается. Предполагается включение в ее состав группы команд, позволяющих моделировать работу устройств и систем, что позволит уменьшить трудоемкость разработки динамических экспертных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[Голенков, 2011] Голенков, В.В. Принципы построения массовой семантической технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н.А Гулякина // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2011): материалы Междунар. научн-техн. конф. (Минск, 10–12 февраля 2011 г.); – Минск: БГУИР, 2011, С. 21–58.

[Джарратано, 2007] Джарратано, Джозеф. Экспертные системы: принципы разработки и программирование, 4-е издание. Пер. с англ. / Джарратано, Джозеф, Райли, Гари ; – М. : ООО "И.Д. Вильямс", 2007.

[Сидоркина, 2011] Сидоркина, И.Г. Системы искусственного интеллекта: учебное пособие / И. Г. Сидоркина; – М.: КНОРУС, 2011.

[Частиков и др., 2003] Разработка экспертных систем. Среда CLIPS /Частиков А.П. [и др.]; - СПб.: БХВ-Петербург, 2003.

EXPERT SYSTEMS DEVELOPMENT IN THE MULTI STUDIO ENVIRONMENT

Kataev V.A.

LtD "Perm Scientific Industrial Instrument-Making Company", Perm, Russian Federation
Bravo55555@yandex.ru

This paper includes short summary of static expert systems development in Multi Studio (MS) Environment. Multi Studio is Russian development. Environment is based on universal language Multi and semantic Multi-networks. Multi-network is combination (symbiosis) of syntactic and semantic networks. Results of the comparison Multi Studio with known environment CLIPS are presented.

Key words: CLIPS, expert system, Multi Studio, semantic network.

INTRODUCTION

Expert systems (ES) belong to class of intelligence systems (IS). IS designing is labor intensive and long. We consider that the main line of IS development is IS designing language improvement and development of effective environment of this language implementation.

Multi Studio system attempts to eliminate this problem by implementing a simple universal high-level language Multi.

Multi main properties are: high-level (Multi language is close to natural language), simplicity, universality, modularity, context using, ability to manipulate network structures objects (network structures are the most universal).

MAIN PART

The main part contains:

1. Description of Multi Studio Environment. MS is a universal multiple-windows interactive Multi language interpreter. Multi language is a high-level command language. Multi belongs to the functional paradigm and has prefix notation.

2. The MS knowledge representation model. System memory ("electronic brain") consists of a cells ("neuros") set. Each "neuro" contains an object (e.g., a number) and its syntax connection to other "neuros". In general, "neuro" is semantic network node. Knowledge representation frame model and production model is a special case of a semantic network. Knowledge bases and programs have similar description in Multi language. They are in the same network.

3. Expert systems development in the Multi Studio Environment. Set of language commands enables to develop in MS Environment static ES. The core command is called "menu" and has the structure: menu.("Question" "answer 1" (action 1) "answer 2" (action 2) ...). The command opens a window, which shows the question, answers (basic buttons) and two additional buttons: the "back" and the "end". When you click on the basic button, the relevant action is executed, e.g. moving to the next command "menu". The "back" returns to the previous menu command. The "end" finishes ES.

ES in Multi Studio is a network of commands "menu" and its components. The command enables forward and backward chaining.

4. Comparison of Multi Studio and CLIPS. Comparison of MS and CLIPS was performed by the example of test ES engine troubleshooting. This ES was taken from developers of system CLIPS.

Results:

The program text on Multi is four times shorter than the program text on CLIPS.

The Multi program uses five operators instead of thirty in the CLIPS program.

The program in CLIPS doesn't have returns to previous questions.

ES development in CLIPS is times more difficult than in Multi Studio.

Multi Studio system is a universal system. It enables other calculations. CLIPS is a specialized system for ES development.

CONCLUSION

Multi Studio system enables to reduce static ES development time and complicity. The additional group of commands will be included in Multi language. This enables simulating devises and systems work and developing dynamic ES.