



# OSTIS-2014

(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

## ОПЫТ ЭКСПЕРИМЕНТАЛЬНОЙ РАЗРАБОТКИ ПРОГРАММНОЙ СРЕДЫ С БАЗОЙ ЗНАНИЙ НА СЕМАНТИЧЕСКИХ СЕТЯХ

Катаев В.А.

*ООО «Центр интеллектуальных технологий»,  
г. Пермь, Российская Федерация*

**bravo666666@yandex.ru**

В работе содержится краткое описание программной среды Multi Studio, которая реализует командный язык Multi на универсальных семантических Multi-сетях.

При разработке MS в качестве образца использовалась информационная среда человека, которая базируется на естественном языке и включает конечный синтаксис, бесконечную расширяющуюся Базу знаний человека и хранилище этой Базы – мозг, состоящий из нейронов и связей между ними.

Аналогично Multi Studio включает высокоуровневый Язык знаний Multi, Базу знаний на семантических Multi-сетях и Машину работы с Базой.

**Ключевые слова:** семантические сети, язык знаний, база знаний, интеллектуальная система, Multi Studio.

### Введение

Потребности в интеллектуальных системах различного назначения стремительно растут. Необходима разработка универсальной технологии создания указанных систем, базирующейся на универсальном языке супервысокого уровня и унифицированных информационных структурах.

Разработка среды Multi Studio (MS) является попыткой экспериментально проверить некоторые идеи, ведущие к решению указанной проблемы.

В этом же направлении весьма масштабно работает Белорусский государственный институт информатики и радиоэлектроники (проект OSTIS, руководитель проекта Голенков В.В.).

В его статьях [Голенков,2012] и [Голенков,2013] поставлена задача разработки Метасистемы IMS.OSTIS, предназначенная для комплексной (в том числе организационной) поддержки проектирования интеллектуальных систем.

### 1. Основные принципы разработки Среды

При разработке MS в качестве образца использовалась информационная среда человека, которая базируется на естественном языке и включает конечный синтаксис, бесконечную расширяющуюся Базу знаний человека и хранилище этой Базы – мозг, состоящий из нейронов и связей между ними.

#### 1.1. Язык знаний

Язык знаний должен быть:

- максимально простым;
- формально однозначным в границах заданных контекстов.

#### 1.2. База знаний

База знаний Среды – это хранилище знаний в электронной памяти, обладающее следующими основными свойствами:

- на физическом уровне – это набор файлов, состоящих из записей (ячеек) весьма сложной структуры. Запись – малый аналог нейрона;
- на логическом уровне – это набор абстрактных узлов с однородной унифицированной структурой, в которых могут располагаться любые одиночные информационные объекты (слова) Языка Среды (числа, даты, команды Языка, имена, отношения ... );
- соединение одиночных информационных объектов в некоторые информационные агрегаты по правилам базового синтаксиса порождает однородные синтаксические сети;
- в сетях объекты-узлы связываются друг с другом направленными переходами, которые формально представляют «родственные» отношения между узлами: «предок-потомок» и «потомок-потомок»;
- синтаксические сети являются основой для семантических сетей, которые представляют собой

различные неоднородные информационные агрегаты: одиночные показатели, списки, множества, матрицы, кортежи, таблицы, деревья, фреймы, правила, онтологии, базы данных, базы знаний, программы;

- семантические сети являются фрагментами единой Базы знаний.

### 1.3. Машина

Машина работы с Базой знаний состоит из набора программ.

Машина имеет 2 основных цикла работы (возможно разделенных во времени):

- перевод (компиляция) текстов программ и данных в унифицированные внутренние структуры и запись их в Базу знаний;
- чтение программ (и данных) из Базы и их исполнение (интерпретация).

Внутренние интерфейсы между локальными и распределенными фрагментами Баз знаний обеспечивает Язык знаний.

Внешние интерфейсы:

- интерфейс с непосредственным пользователем Среды (обозначения команд Языка знаний, комментарии программ в процессе работы, тексты в Базе знаний):

- осуществляется на текстовых фрагментах естественного языка (Язык пользователя);

- в качестве Языка пользователя может применяться русский язык, английский и другие;

- при этом Машина должна обеспечивать конкретного пользователя простыми средствами создания своих национальных синонимов стандартных обозначений Языка пользователя;

- интерфейс с внешними программными системами, работающими на иных языках и структурах:

- минимально требуется перевод внешних баз знаний в Базу Среды и/или обратно;

- в принципе, программы указанных переводов можно делать на любой стороне. Предпочтительней - на стороне Среды.

Непосредственное использование Среды в качестве Вычислителя (Решателя) производится преимущественно с использованием Языка знаний. Информационные системы, созданные на базе Среды, напротив, общаются с оператором в основном на Языке пользователя.

## 2. Среда Multi Studio

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

Сеанс работы с MS называется беседой и состоит из серии заданий. Задание на языке Multi (в текстовом варианте) записывается в окно ввода.

Результат выполнения задания выдается, как правило, в окно вывода.

Язык пользователя – русский язык. В эксперименте опробован английский.

Язык знаний - Multi – это высокоуровневый командный (алгоритмический) язык с функциональной парадигмой и префиксной формой записи.

### 2.1. Базовый синтаксис Языка знаний

Описание базового синтаксиса Языка (во внешнем текстовом варианте):

- multi-текст задания – это непустой список выражений, разделенных пробелами;

- выражение [семья] – это предок [объект] и потомство, которого может не быть;

- потомство – это список выражений, заключенный в круглые скобки. Таким образом, выражения могут быть многоуровневыми;

- в конце выражения может присутствовать символ точка с запятой, который явно обозначает конец выражения и позволяет не указывать последние закрывающие скобки выражения;

- объект: первичный (терминальный) или вторичный (составной);

- к первичным объектам относятся: число (-5.667), литерал («Расчет плана»), дата (25.12.2014), имя (Расчет\_плана), понятия ('человек', 'любит'), команды языка;

- команды Языка: Сумма (+), музыкальная студия (MIDI), другие (более 500); Команды представлены в Базе номерами. Обозначения команд представлены в таблицах команд вне Базы. Пользователь может ввести в таблицу свои национальные обозначения команд в качестве синонимов к стандарту;

- к составным объектам относится: пара (25!метров), тройка (2!40!метров), другие;

- пара состоит из двух первичных объектов, разделенных символом ! :

- +!доллар [вычислить сумму долларов];
- 3!4 [музыкальный размер «вальс»].

Пример Multi-текста из двух выражений:

- P# ("сумма="+(X 123 ; X ( 5000) (1)
- где P# [вывод на экран объектов потомства через пробел];
- вывод результата: 'сумма= 5123'.

### 2.2. Multi-сеть представления знаний

#### 2.2.1. Бункер

На физическом уровне База знаний среды MS, называемая Бункером, является файлом. Первоначально в Бункере находится единственная синтаксическая сеть Куча в виде списка, в котором находятся все пустые узлы Бункера.

Бункер подразделяется на множество Сфер. Каждая Сфера имеет свой каталог имен. В каталоге Сферы не допускается дублирование имен. Но в

разных Сферах можно иметь имена-омонимы. Благодаря этому Сферу можно использовать для хранения знаний определенной предметной области, не заботясь о совпадении обозначений объектов с разной семантикой из соседних областей (например, класс в объектном программировании, в образовании и в политике).

### 2.2.2. Компиляция

При запуске задания исходный Multi-текст компилируется в синтаксическую сеть задания и записывается в Бункер.

На первом этапе для каждого выражения создается отдельная синтаксическая сеть.

Копия объекта (или ссылка на него) помещается в свободный узел. Между узлами устанавливаются переходы:

- открывающая скобка (исходного текста) – переход "предок→потомок";
- пробел между объектами – "потомок→потомок";
- закрывающая скобка – "потомок→пусто".

Каждый переход между двумя узлами предусматривает встречную запись адресов связываемых узлов. Кроме того узлы содержат дополнительные поля, которые используются при работе с узлами. Например, для накопления суммы командой +.

На втором этапе сети выражений связываются в общую синтаксическую сеть задания:

- сети именованных выражений (у которых предок - имя) связываются переходами "предок→потомок";
- Сети неименованных выражений - переходами «потомок→потомок».

При отсутствии в сети синтаксических ошибок Машина запускает задание.

Пример: на рисунке 1 представлен графический вид сети задания (1). Переходы между узлами представлены непрерывными стрелками. Исполнение – пунктирными стрелками.

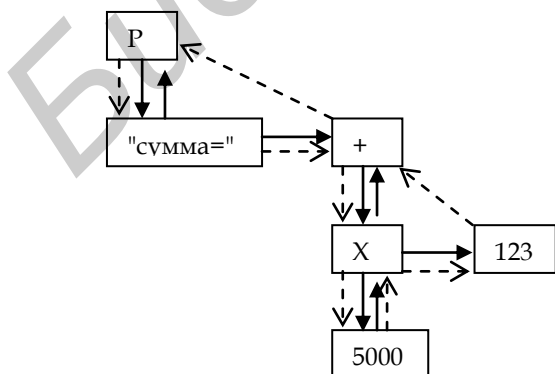


Рисунок.1 – Сеть задания (1) и маршрут исполнения

### 2.2.3. Исполнение задания

Машина производит выполнение задания, начиная с первого узла задания путем перехода от узла к узлу преимущественно по принципу "сверху вниз [предок→потомок] и слева направо [потомок→потомок]". Некоторые команды (.. [цикл]) могут изменять маршрут исполнения.

Синтаксически правильное задание может быть прервано из-за семантических ошибок.

### 2.2.4. Семантика

Базовый синтаксис локален и предельно прост. Его правила распространяются только на выражения одного уровня. Семантика глобальна.

Семантика выражений анализируется не в фазе компиляции, а в процессе их исполнения. Причина: в статической фазе компиляции нет динамических контекстов, необходимых семантике.

По времени и области действия (контексту) различаются семантики Бункера, беседы, задания и выражения. Каждая из 500 команд Языка имеет свою семантику и выступает в качестве параллельно-последовательного агента вместе с другими при выполнении задания.

Имеются глобальные команды, которые действуют на все уровни выражения любого задания через глобальные параметры. Например, команда *сфера.*, которая задает идентификатор текущей сферы Бункера. Семантическая структура выражения этой команды: *сфера.(имя\_сферы)*.

Первоначальные значения глобальных параметров задает Машина. Она же сохраняет последние изменения.

Большинство команд являются локальными и действуют только на потомков своего выражения.

Например, команда + имеет следующую семантику: + (*потомство*). Область ее действия - все потомки-числа всех уровней, являющиеся константами или полученными в результате вычислений другими командами. Другие объекты для нее нейтральны. Если среди нейтралов встречается команда, она прерывает действие + и работает в пределах своего потомства, а по окончании возвращает вычисленное значение и управление старшей команде.

Пример:

P#( + ( 111 P# («Привет!») +( 5 5 ( 10 ) ) ) )  
 Результат выводится на экран: Привет! 131

Многоагентная работа команд Среды позволяет иметь в одном задании программу оркестровки песни, в текст которой вставлены команды многоуровневой сортировки и вывода таблиц базы данных, а также предикаты универсума Эмбрана, представленные на рис. 4.3 в [Осипов,2012] и все это выполнить: создать и запустить песню, выдать отсортированную таблицу и узнать, имеет ли Петя собаку. Проверено: работает.

Язык Multi обладает несколькими полезными свойствами, облегчающими семантическую отладку программ:

- Повышенный уровень Языка.
- Простейший синтаксис.
- Простая всепроникающая декомпозиция

Multi-программ посредством имен. При этом отладку отдельных поименованных фрагментов программы можно вести последовательно и/или даже независимо параллельно. Отсюда существенно уменьшается общая сложность и сроки отладки программ.

- Наличие команды *test*. [пошаговая динамическая диагностика действий программы с любого узла исполнительного маршрута сети].

- Наличие команды *##* [вывод статической структуры сети любого узла любой сложности с заданным числом уровней потомков].

- Визуальное хождение пользователя по Базе знаний в окне Среды.

## Заключение

Получен опыт применения Среды для создания различных интеллектуальных систем: экспертная система промышленного назначения, учебный программный имитатор управления самоходной артустановкой, виртуальная звуковая студия с использованием музыкального расширения языка Multi вместо классической нотации. А также имеется опыт использования среды при обучении школьников информатике.

На основе опыта разработок Multi Studio, OSTIS, ИМПЗ [Загорюлько, 2013] и некоторых других рекомендуется открыть международный проект по развитию инструментальных средств разработки интеллектуальных систем.

## Библиографический список

[Голенков, 2012] Голенков, В.В. Графодинамические модели параллельной обработки знаний: принципы построения, реализации и проектирования / В.В. Голенков, Н.А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2012): материалы Междунар. научн-техн. конф. (Минск, 16–18 февраля 2012 г.); – Минск: БГУИР, 2012, С. 23–52.

[Голенков, 2013] Голенков, В.В. Открытый проект, направленный на создание технологии компонентного проектирования интеллектуальных систем / В.В. Голенков, Н.А. Гулякина // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): материалы Междунар. научн-техн. конф. (Минск, 16–18 февраля 2013 г.); – Минск: БГУИР, 2013, С. 55-78.

[Загорюлько, 2013] Загорюлько Ю.А. Технологии разработки интеллектуальных систем, основанные на интегрированной модели представления знаний // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2013): материалы Междунар. научн-техн. конф. (Минск, 16–18 февраля 2013 г.); – Минск: БГУИР, 2013, С. 31-42.

[Осипов, 2012] Осипов Г.С. Лекции по искусственному интеллекту. Изд. 2-е, испр. и доп. М.: Книжный дом «ЛИБРОКОМ», 2013. -272с. (Науки об искусственном.)

## EXPERIENCE OF EXPERIMENTAL DEVELOPMENT OF SOFTWARE ENVIRONMENT WITH THE KNOWLEDGE BASE ON SEMANTIC NETWORKS

Kataev V.A.

*Center of Intelligent Technologies, Ltd, Perm,  
Russian Federation*

**Bravo666666@yandex.ru**

The paper presents a brief description of the Multi Studio environment which realizes the Multi command language on universal semantic Multi-networks.

## Introduction

Needs for intelligent systems of various purpose grow promptly. That is why it is necessary to develop a universal technology of creation of these systems, based on a universal language of superhigh level and unified information structures.

Development of Multi Studio (MS) is an attempt to test experimentally some of ideas leading directly to the solution of the specified problem.

## Main Part

When developing MS, as an example we used the information environment of a human based on a natural language; this environment includes finite-state syntax, the infinite extending Knowledge base of a human and a human brain (consisting of neurons and connections between them) as a repository of this Base.

Similar to this, our Multi Studio includes Multi (the high-level Knowledge Language), the Knowledge base on semantic Multi-networks and the Software to work with the Base.

The Multi-network consists of uniform abstract knots which are connected by directed transitions: "ancestor-descendant", "descendant-descendant". Various information units can be based in Multi-networks, such as single units, lists, sets, matrixes, tuples, tables, trees, frames, rules, ontologies, databases, knowledge bases, programs.

All those features of the environment along with simplicity of the source language allow a user to apply Multi Studio both for its direct purposes as a "calculator", and as a tool environment for developing intelligent information systems.

We have an experimental experience of application of the environment for developing an industrial-purpose expert system, training program simulator of self-propelled assault gun operation, virtual sound studio.

## Conclusion

On the basis of experience of development of MS, OSTIS and the others, it is recommended to start the international project on elaboration of Tools for development of intelligent systems.