

# СЕМАНТИЧЕСКИЕ СРЕДСТВА ВИЗУАЛИЗАЦИИ ПРОЦЕССА ОТЛАДКИ ПРОГРАММ ОБРАБОТКИ ЗНАНИЙ

М. О. Данилов, А. С. Бельчиков, А. Л. Козел

Кафедра интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь

E-mail: maxdanilovsvarog@gmail.com, thecubeismagic@gmail.com, belchikov1996@gmail.com

*Технология OSTIS - открытая семантическая технология разработки интеллектуальных систем. Для описания способов решения задач и поведения агентов над общей графодинамической памятью предполагается использовать графовые языки программирования[1-2]. Как и все языки программирования, графовые языки должны иметь средства отладки написанных на них программ.*

## ВВЕДЕНИЕ

SC-код - базовый язык внутреннего смыслового представления знаний в технологии OSTIS и в создаваемых с её помощью системах. В качестве базового языка написания программ обработки знаний предлагается язык SCP[2]. Это графовый язык процедурного программирования, предназначенный для эффективной обработки семантических сетей с теоретико-множественной интерпретацией.

Процесс отладки таких программ коренным образом отличается от отладки обычных программ. В первую очередь это связано с тем, что в отличие от обычной программы, такая программа работает с динамической семантической сетью, и сама является элементом этой сети. Поскольку программы обработки знаний работают с общей памятью и являются её частью, то одна программа может изменить состояние базы знаний таким образом, что, например, может привести к непредвиденному результату работы второй программы. По этой причине существующие инструменты отладки не удовлетворяют потребностям процесса отладки программ обработки знаний. Для наглядности процессов, происходящих в памяти, разработчику требуется удобный способ доступа к состоянию памяти в любой момент выполнения программы. Решить эту проблему позволяет создание семантических средств визуализации процесса отладки.

### I. ОСНОВНОЙ ПОДХОД К ОТЛАДКЕ ПРОГРАММ, ОБРАБАТЫВАЮЩИХ ЗНАНИЯ

В первую очередь, в режиме отладки пользователь имеет возможность найти в памяти остановленный SCP-процесс.

Далее средствами отладки процесс визуализируется в окне отладчика. Процесс является узлом графосемантической памяти. На основе этого строится граф. Этот граф содержит вершины-операторы и дуги, связи между вершинами, которые являются переходами. Красным цветом отмечается начальный оператор, зеленым цветом - конечные операторы, жёлтым цветом - те-

кущий оператор. Черным обведятся операторы, которые являются точками останова программы.

В ходе отладки пользователь имеет возможность перейти к следующему оператору. При нажатии на кнопку перехода текущий оператор выполняется, и процесс выполнения программы останавливается на следующем операторе.

Так же пользователь имеет возможность создать произвольную точку останова и продолжить работу программы до её достижения или до окончания.

Для максимально удобного и продуктивного изучения работы программы разработчик имеет возможность просмотра семантической окрестности оператора. Так как во время выполнения программы любой оператор не перестаёт быть частью графосемантической базы знаний, у разработчика есть возможность не только получить семантическую окрестность оператора, но и сразу же увидеть все связанные с этим оператором или программой данные. Это позволяет упростить механизм нахождения интересующих узлов в памяти в сравнении с использованием протоколирования, а так же значительно увеличивает продуктивность разработки программ, ориентированных на обработку знаний. Для просмотра семантической окрестности SCP-программы и ее операторов были разработаны соответствующие агенты: агент поиска семантической окрестности SCP-программы, агент поиска семантической окрестности оператора.

### II. ПРОБЛЕМА ВИЗУАЛИЗАЦИИ ГРАФА ПРОГРАММЫ ПРИ ОТЛАДКЕ

Существенной проблемой при разработке средств отладки программ ориентированных на обработку знаний стало отображение графа программы. Удобное отображение графа программы максимально информативно для разработчика, поэтому эта проблема существенна. Решение этой проблемы должно обеспечить читабельность и удобство работы разработчика с программой.

Можно выделить следующие способы отображения графов[3–5]:

- произвольное;
- сеточное;
- ортогональное;
- послонные методы;
- дуговая диаграмма;
- круговой.

Сеточный метод предполагает, что вершины, а также все точки пересечения и сгибы ребер имеют целочисленные координаты, т. е. находятся в узлах координатной сетки, образованной прямыми, параллельными координатным осям и пересекающимися их в точках с целочисленными координатами

Послойные методы располагают вершины на нескольких горизонтальных линиях или слоях, и дугах, направленных в общем вниз.

Ортогональный метод состоит из 3 фаз. В первой фазе подсчитываются подсчитываются пересечения ребер. Во второй фазе вычисляются отводы на чертеже, на третьем этапе определяются окончательные координаты. Данный алгоритм располагает дуги вертикально или горизонтально, параллельно координатным осям, минимизируя скрещивания дуг путем замены их на вершины. ортогональный метод хорошо подходит для средне разреженных графов.

Метод дуговой диаграммы предполагает расположить вершины на одной линии в Евклидовом пространстве и дугами, нарисованными в одной из полуплоскостей ограниченной этой линией.

Круговой метод располагает вершины графа на окружности, осторожно выбирая порядок расположения вершин, чтобы уменьшить пересечения ребер и располагать смежные вершины близко друг к другу. Дуги могут быть нарисованы или как хорды круга, или как дуги внутри или снаружи круга. В некоторых случаях могут использоваться несколько кругов.

Для визуализации программы обработки знаний были приняты следующие соглашения:

- вершины графа не должны лежать на ребрах, концами которых они не являются;
- вершины не должны находиться слишком близко;

- углы между пересекающимися ребрами больше;
- число пересечений ребер должно быть меньше;
- площадь укладки меньше;
- читабельность.

В текущей версии платформа OSTIS предоставляет возможность только произвольного отображения графа, и это мало чем помогает разработчику в отладке.

С целью добиться максимальной пользы для разработчика при разработке средств визуализации процесса отладки программ обработки знаний было решено комбинировать метод дуговой диаграммы отображение графов.

### III. ВЫВОД

Главным преимуществом использования средств визуализации процесса отладки программ обработки знаний является ощутимое увеличение продуктивности разработчика. Их использование существенно облегчает процесс разработки интеллектуальных систем и подталкивает к разработке различных интеллектуальных средств отладки и разработки, позволяющих прямо во время выполнения программы изменять её тело, подключать и отключать различные модули и компоненты библиотек, изменять данные(знания), с которыми она работает. Всё это позволит максимально быстро и качественно разрабатывать наиболее эффективные машины обработки знаний интеллектуальных систем.

1. Открытые семантические технологии проектирования интеллектуальных систем (OSTIS-2016): материалы VI междунар. науч.-техн. конф. (Минск, 18–20 февраля 2016 года)/ редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2016. – 596 с.
2. Шункевич Д.В. Принципы проектирования и интерпретации программ, ориентированных на обработку семантических сетей /Д.В. Шункевич; Информационные технологии и системы (ИТС-2013): материалы Междунар. конф. , Минск: БГУИР, 2013. – 117 с.
3. Saaty, Thomas L. (1964), "The minimum number of intersections in complete graphs Proc. Natl. Acad. Sci. U.S.A., 52: 688–690
4. Purchase, H. C.; Cohen, R. F.; James, M. I. (1997), "An experimental study of the basis for graph drawing algorithms"(PDF), Journal of Experimental Algorithmics, 2, Article 4
5. Di Battista et al. (1994), Chapter 5, "Flow and Orthogonal Drawings pp. 137–170;