OSTIS-2012
(Open Semantic Technologies for Intelligent Systems)

УДК 004.822:514

ЯЗЫКИ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ, ОРИЕНТИРОВАННЫЕ НА ОБРАБОТКУ СЕМАНТИЧЕСКИХ СЕТЕЙ

Гулякина Н.А., Пивоварчик О.В., Лазуркин Д.А.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

guliakina@bsuir.by

pivovarchyk@tut.by

lazurkin@bsuir.by

В работе рассматриваются языки и технологии программирования, ориентированные на обработку семантических сетей.

Ключевые слова: интеллектуальная help-система, семантическая теория программ, язык программирования, база знаний, графовые языки программирования.

Введение

Интеллектуализация является одним из основных направлений развития современных компьютерных систем. Для эффективного использования интеллектуальных технологий разработчики должны владеть методами представления знаний, методами моделирования баз знаний, технологиями проектирования программ, обрабатывающих знания. Высокая сложность и уникальность разработки интеллектуальных систем предъявляет высокие требования к квалификации их разработчиков. Поэтому важной задачей является повышение качества и сокращение сроков освоения этих технологий. Одним из способов решения этой задачи является создание программных средств поддержки разработки интеллектуальных систем, которые решают задачи информационного обслуживания по проектированию программного обеспечения и аккумулируют опыт квалифицированных программистов и преподавателей. Такие программные средства будем называть интеллектуальными help-системами. Интеллектуальные help-системы позволяют повысить уровень интеллектуальной информационной поддержки разработчиков программ. Наличие help-системы в составе инструментального средства проектирования программ позволит снизить начальные квалификационные требования к разработкам программ и повысить количество проектируемых интеллектуальных информационных систем.

В данной статье описана интеллектуальная help-система по разработке программ, ориентированных на обработку семантических сеией. В качестве языка программирования, ориентированного на обработку семантических сетей, в данной статье рассматривается язык SCP (Semantic Code Programming) [Голенков, В.В. и др., 2001b].

1. Языки программирования, ориентированные на обработку знаний

Языки программирования, ориентированные на обработку знаний, существенно отличаются от языков, ориентированных на обработку традиционных структур данных. Особенностью таких языков является:

- Использование специальных языков для представления обрабатываемых знаний;
- наличие гибких средств управления знаниями или базами знаний.

Среди языков, ориентированных на обработку знаний, выделяются языки программирования, ориентированные на обработку семантических сетей. Под семантической сетью понимается графовая структура, вершины и дуги которой обладают определенной семантической интерпретацией. [Голенков, В.В. и др., 2001a] Переработка знаний, представленных в виде семантических сетей, сводится к изменению конфигурации сети. Языки, тексты которых являются семантическими сетями, называются семантическими графовыми языками.

Примерами графовых языков представления и обработки знаний являются:

- сетевой язык представления фреймов [Поспелов Д.А. 1990];
- язык растущих пирамидальных сетей [Гладун В.П.,1987];
- графовый язык представления информации в памяти вегетативной машины [Борщев В.Б, 1990];
- графовый язык представления информации в памяти абстрактной сетевой машины [Степанов А.М., 1981];
- графовые логические языки [Вагин В.Н., 1989];
- волновые языки, обеспечивающие решение задач навигации на семантических сетях;
- графовые языки представления различным образом устроенных продукционных программ, ориентированных на переработку семантических сетей [Кузнецов В.Е., 1989];
- язык передачи маркеров и др.

К достоинствам семантических графовых языков представления знаний, по сравнению с другими способами представления знаний, относятся:

- компактность, обусловленная тем, что в отличие от символьного текста в семантической сети знак каждого объекта или понятия описываемой предметной области присутствует только в одном экземпляре и состоит из одного элемента;
- ассоциативность, заключающаяся в существовании простых процедур поиска элементов семантической сети, связанных заданным образом с заданными элементами;
- наличие простой возможности введения метаинформации в семантическую сеть путем простого наращивания исходной семантической сети метасетью без какого-либо изменения исходной семантической сети;
- возможность рассмотрения описываемых предметных областей одновременно на неограниченном числе уровней детализации;
- приспособленность к поддержке структур любого вида, и в частности к поддержке сложноструктурированных знаний;
- приспособленность к интеграции самых различных специализированных языков и самых различных моделей представления знаний;
- приспособленность к представлению различного рода лингвистических знаний (о синтаксисе, о семантике, о прагматике естественных языков), что делает эффективным использование семантических графовых языков для создания естественно-языковых интерфейсных подсистем в интеллектуальных системах;
- приспособленность к параллельной асинхронной переработке знаний, что делает эффективным использование семантических графовых языков для создания интеллектуальных систем, поддерживающих сложноструктурированные знания и сложные логические операции.

Для представления и обработки семантических сетей в проекте OSTIS (Open Semantic Technology for Intelligent Systems) [Электронный ресурс] используется семейство графовых языков, текстами которых являются sc-конструкции. Такие языки называются sc-языками. Базовым языком для построения sc-языков является SC-код. [Голенков В.В. и др., 2001a] С помощью SC-кода можно представлять любую информацию, в том числе и нечисловую, описывать сложноструктурируемые предметные области. На базе SC-кода разработано множество языков представления знаний, ориентированных на описание определенных предметных областей.

Хранимые в памяти интеллектуальной системы процедурные знания осуществляют описание методов решения различных классов задач и, следовательно обеспечивают, с одной стороны, существенное повышение производительности интеллектуальной системы (если в состав процедурных знаний будут включены описания методов решения наиболее часто решаемых задач) и, с другой стороны, обеспечивают неограниченное расширение и модифицируемость системы операций переработки знаний.

К числу процедурных знаний относятся:

- всевозможные команды, каждая из которых описывает некоторое действие, считающееся элементарным в рамках соответствующей абстрактной машины и направленное либо на преобразование информации, хранимой в памяти этой машины, либо на преобразование ее внешней среды;
- всевозможные инициированные команды (операторы, инструкции), представляющие собой описание элементарных действий, подлежащих выполнению;
- всевозможные процедурные программы, каждая из которых представляет собой систему команд, инициируемых по тем или иным правилам;
- информационные конструкции, каждая из которых представляет собой описание текущего состояния процесса реализации некоторой процедурной программы.

Без активного использования в интеллектуальных системах различных программ (т.е. программных знаний о методах решения конкретных классов задач) практически невозможно создавать эффективные прикладные интеллектуальные системы. Таким образом, возникает проблема создания таких моделей переработки знаний и таких моделей реализации программ, которые бы легко интегрировались в одну модель, в рамках которой некоторые фрагменты знаний одновременно рассматривались бы как данные для соответствующих программ.

Интеграция формальных моделей реализации хранимых программ с формальными моделями

переработки знаний обеспечивает повышение производительности интеллектуальных систем (если в базу знаний включить программы, соответствующие часто решаемым задачам), а также обеспечивает открытый характер и легкую модифицируемость системы операций переработки знаний (для этого в базу знаний необходимо включить программы, которые описывают указанные операции).

Реализацию произвольной графодинамической формальной модели условно можно разбить на три этапа:

- приведение реализуемой графодинамической модели к каноническому виду, т.е. построение эквивалентной sc-модели;
- запись микропрограмм операций (правил вывода) для канонического вида реализуемой формальной модели (sc-модели) на некотором языке программирования;
- техническая разработка интерпретатора или компилятора указанного языка программирования, на котором записываются микропрограммы операций реализуемых формальных моделей.

Для обработки знаний, представленных с помощью sc-языков, существуют языки программирования:

- SCP (Semantic Code Programming) – базовый процедурный язык программирования;
- SCPH (Semantic Code Programming High-level) – процедурный язык программирования высокого уровня;
- функциональный язык программирования;
- логические языки программирования на базе языка SCL.

Данные языки программирования разработаны на базе SC-кода. Таким образом, они также являются sc-языками. Следовательно, программы и перерабатываемые ими данные, являются sc-конструкциями. Построение языков представления знаний и языков обработки знаний по одному принципу обеспечивает представление данных и программ в виде sc-конструкций, что позволяет легко интегрировать различные виды информации и хранить в одной базе знаний декларативные и процедурные знания.

2. Технология проектирования программ, ориентированных на обработку семантических сетей

Центральной проблемой любой эффективной технологии проектирования систем является то, как должна быть устроена система, чтобы ее легче было разрабатывать, модифицировать, пополнять и сопровождать. Такая технология должна быть ориентирована не столько на разработку определенного класса систем, сколько на их постоянное совершенствование.

Технология должна включать в себя:

- теорию проектируемых программ – уточнение того, как устроены проектируемые программы того класса, на которые ориентирована данная технология;
- структурированную библиотеку типовых многократно используемых компонентов проектируемых программ;
- инструментальные средства (средства автоматизации) проектирования;
- методику проектирования;
- методику обучения проектированию;
- интеллектуальную help-систему для информационного обслуживания и обучения проектировщиков;
- инфраструктуру, обеспечивающую организацию проектирования.

3. Язык программирования SCP

SCP (Semantic Code Programming) [Голенков, В.В. и др., 2001b] – это базовый язык программирования, ориентированный на обработку однородных семантических сетей, имеющих базовую теоретико-множественную интерпретацию. Для представления таких семантических сетей используется SC-код. Язык SCP является sc-языком, т.е. тексты языка (scp-программы) представляются в виде sc-конструкций. Программы языка SCP и перерабатываемые ими информационные структуры (sc-конструкции) могут хранится в одной базе знаний. Тест языка SCP с формальной точки зрения есть определенным образом устроенная sc-конструкция, хранимая в sc-памяти и описывающая некоторую систему параллельных взаимодействующих процессов переработки другой sc-конструкции, хранимой в той же sc-памяти. Особенностями языка SCP являются:

- ориентация на использование структурно перестраиваемой (графодинамической) ассоциативной памяти;
- хорошая приспособленность к переработке нечисловых структур, что обеспечивает описание сложно структурированных предметных областей;
- ориентация на переработку непосредственно семантических сетей;
- возможность описания механизмов решения задач различного уровня сложности;
- возможность через понятие sc-узла интеграции scp-программ с программами, написанными на других языках программирования и описывающими переработку информационных структур;
- высокий потенциал распараллеливания процессов переработки информации в графовой структурно перестраиваемой ассоциативной памяти [Голенков, В.В. и др., 2001b].

Язык SCP обладает мощной типологией операторов, позволяющей производить различные виды операций с информационными структурами, представленными в виде семантических сетей.

Типология операторов языка SCP:

- операторы генерации sc-конструкций;
- операторы удаления sc-конструкций;
- операторы ассоциативного поиска sc-конструкций;
- операторы выборки;
- операторы проверки условий;
- операторы изменения свойств sc-элемента;
- операторы управления scp-процессом;
- операторы вывода.

Операционная семантика языка программирования SCP задается соответствующей абстрактной машиной. Абстрактная машина задается абстрактной памятью, в которой хранятся перерабатываемые конструкции, и множеством операций. Все вспомогательные данные, необходимые абстрактной scp-машине для реализации хранимых в памяти scp-программ, так же представляются в виде sc-конструкций. Абстрактная память является динамической информационной конструкцией. Структура памяти определяется синтаксисом языка, принципы изменения памяти определяются операциями.

Язык SCP разрабатывается и реализуется в нескольких модификациях, имеющих одинаковый набор непосредственно выполняемых операций, но различный синтаксис. Система операций языка SCP описана в [Голенков В.В. и др., 1994a], [Голенков В.В. и др., 1994b]. Реализован интерпретатор, обеспечивающий выполнение всех операций, включенных в текущую версию языка SCP. Для каждой из модификаций языка SCP дополнительно к указанному интерпретатору требуется реализовать конвертор из соответствующей формы представления scp-процедур в форму, поддерживаемую интерпретатором. Средства разработки программ, ориентированных на обработку семантических сетей

Инструментальное средство проектирования программ базового языка программирования, ориентированного на обработку унифицированных семантических сетей, поддерживают представление scp-программ в графической форме (SCg) и в линейной форме (SCs, M4SCP). Инструментальное средство, используемое в настоящее время, включает: редактор scp-программ, средство сборки репозитория исходных текстов, scp-интерпретатор, scp-отладчик с возможностью визуального отображения процесса отладки scp-программ [Лазуркин Д.А. и др., 2010].

Так как SCP является процедурным языком, то основная схема традиционного подхода отладки программ подходит и для создания отладчика scp-программ. Однако следует отметить, что тексты scp-программ представлены в виде семантических сетей и хранятся в ассоциативной памяти, что расширяет возможности отладки таких программ и упрощает сам отладчик по следующим причинам:

- структуру программ, представленных в виде sc-конструкций, легче модифицировать, чем их линейные аналоги;
- ассоциативный доступ к sc-памяти и явное выделение связей между частями scp-программы позволяет отладчику проявлять интеллектуальность без анализа низкоуровневой структуры отлаживаемой программы;
- часть символьной информации (имена программ, имена переменных) хранится в sc-памяти в виде идентификаторов соответствующих sc-элементов;
- упрощается возможность добавления к программе символьной метаинформации.

Процесс, интерпретирующий scp-программу, должен сообщать отладчику о ходе своей работы, поэтому для поддержки возможности отладки scp-программ необходимо расширить модель scp-процесса [Голенков, В.В. и др., 2001], [Голенков, В.В. и др., 2001b], [Электронный ресурс].

Трассировщик scp-процесса – это программа, которая сопоставлена данному scp-процессу и получает сведения о ходе его выполнения. На данном этапе разработки будем считать, что трассировщик является микропрограммой , т.е. его текст написан на языке внешнем по отношению к sc-памяти. Расширим связку, которая обозначает scp-процесс дополнительным компонентом с атрибутом *ptrace_*, который является знаком трассировщика. Таким образом, каждому scp-процессу может быть поставлен в соответствие только один трассировщик. Управление и всю необходимую информацию трассировщику передает scp-процесс при возникновении следующих ситуаций:

- осуществилось выполнение одного scp-оператора (только в случае пошаговой трассировки);
- в результате интерпретации scp-оператора произошла ошибка;
- scp-процесс создал процесс-потомок;
- scp-процесс завершился.

Отладка является частным случаем трассировки scp-процесса, поэтому создана специальная микропрограмма трассировщика. Она имеет доступ ко всем структурам scp-процесса и при необходимости может приостановить scp-процесс, исключив его из очереди планировщика. Эта же микропрограмма получает следующие сообщения управления от пользовательского интерфейса:

- продолжить выполнение текущего отлаживаемого scp-процесса, который приостановлен («продолжить исполнение»);
- включить режим пошаговой трассировки и перейти к следующему scp-оператору текущей scp-программы («шаг трассировки»);
- перейти к начальному scp-оператору scp-программы, которую вызывает текущий scp-оператор («шаг во внутрь»).

При создании дочернего scp-процесса для него устанавливается трассировщик родительского scp-процесса, если родительский scp-процесс находится в режиме трассировки.

Предложенный механизм низкоуровневой отладки scp-программ обеспечивает основу для организации визуальной отладки с отображением графодинамического процесса преобразования sc-памяти. Инструментальное средство является интеллектуальной системой и построено с использованием открытой семантической технологии.

4. Интеллектуальная help-система по проектированию программ, ориентированных на обработку семантических сетей

Интеллектуальная help-система является консультантом в области технологии проектирования программ, ориентированных на обработку семантических сетей. Для проектирования интеллектуальной help-системы используется открытая семантическая технология компонентного проектирования интеллектуальных систем. В основе семантической технологии проектирования интеллектуальных систем лежит семантическая модель представления знаний, которая использует семантическую сеть с базовой теоретико-множественной интерпретацией и представление знаний в sc-коде. Для представления знаний в базе знаний help-системы используется sc-код. [Голенков В.В. и др., 2001а] Таким образом, help-система является sc-системой.

Help-система состоит из следующих компонентов:

- интеллектуальная help-система по семантической теории программ;
- интеллектуальная help-система по библиотеке программ, ориентированных на обработку семантических сетей;
- интеллектуальная help-система по комплексу инструментальных средств проектирования программ, ориентированных на обработку семантических сетей;
- интеллектуальная help-система по методике проектирования программ, ориентированных на обработку семантических сетей;
- интеллектуальная help-система по методике обучения проектированию программ, ориентированных на обработку семантических сетей.

Каждый компонент содержит знания из соответствующей области технологии проектирования программ. В соответствии с открытой семантической технологией каждый компонент включает:

- справочную подсистему;
- подсистему мониторинга и анализа деятельности разработчика программ;
- подсистему управления обучением.

Каждая из подсистем взаимодействует с другими подсистемами, а также может функционировать автономно. Разработка каждой подсистемы состоит из проектирования:

- базы знаний;
- машины обработки знаний;
- пользовательского интерфейса.

Одним из основных принципов открытой семантической технологии является унификация интеллектуальных систем. Унификация позволяет не только упростить разработку интеллектуальных систем, но и интегрировать как отдельные компоненты, так и целые системы. Это позволяет реализовывать каждый компонент help-системы и каждую подсистему отдельно и легко интегрировать их. В настоящее время разрабатывается справочная подсистема help-системы по семантической теории программ.

Справочная подсистема является консультантом-экспертом в области семантической теории программ, который может ответить на любой вопрос новичка или опытного пользователя. К функциям справочной подсистемы относятся:

- поиск информации по запросу пользователя, в том числе свободно-конструируемому;
- отображение найденной информации с учетом уровня квалификации пользователя;
- анализ программных текстов и внесение предложений по улучшению их эффективности;
- генерация программных текстов по запросу пользователя.

При проектировании интеллектуальной системы семантическая технология вначале предполагает разработку семантического языка представления вопросов. Разработанный язык определяет способы формулирования запросов к хранящейся в базе знаний информации и формирования ответов по каждому выделенному классу вопросов. Классы вопросов в разрабатываемой справочной системе определяются в соответствии с процедурами (операциями), которые требуются при ответе на них. В справочной подсистеме используются базовые классы семантического языка вопросов, а также выделены специальные классы. Основными классами вопросов являются:

- стандартные вопросы (например, запросы всех известных элементов заданного множества; запросы всех известных множеств, элементом которых является указываемый sc-элемент);
- запросы высказываний, связанных с заданными объектами, обобщенными структурами или высказываниями (например, относится ли scp-оператор *genEl* к классу scp-операторов поиска);
- запросы, связанные с классификацией множеств (например, какие классы scp-операторов являются подклассами операторов поиска);
- запросы, связанные с отношениями (например, привести отношения, заданные на множестве scp-программ);
- запросы, связанные с внешней идентификацией элементов базы знаний и с внешним представлением различных ее фрагментов;

- запросы, связанные с определениями и пояснениями понятий (например, привести определение понятия scp-переменная);
- запросы фрагментов семантических окрестностей элементов баз знаний (например, вывести полную семантическую окрестность понятия scp-оператор);
- запросы, связанные с доказательствами (например, истинным ли является данное высказывание «*scp-оператор genEl генерирует одноэлементную sc-конструкции в базе знаний*»);
- запросы, связанные с программами (например, каким способом можно посчитать мощность множества);
- запросы фрагментов логических спецификаций формальных теорий (например, как выглядит теоретико-множественная система основных классов объектов семантической теории программ).

При анализе ответов на вопросы выделены ключевые понятия, которые легли в основу базы знаний, и определены отношения, для установления связей между ними. Также, для каждого понятия в базе знаний заданы его теоретико-множественные свойства. К теоретико-множественным свойствам можно отнести: пояснение и определение понятия, разбиение на более частные понятия, включение в более общие понятия, синонимия понятия, пересечение и объединение с другими понятиями, основные утверждения о понятии, примеры понятия и др.

В базе знаний справочной подсистемы интеллектуальной help-системы по семантической теории программ представлена информация о семантической теории программ, описаны синтаксис и денотационная семантика базового языка программирования, ориентированного на обработку унифицированных семантических сетей (языка программирования SCP), абстрактная машина, интерпретирующая базовый язык программирования.

Для описания семантической теории программ в виде семантической сети представлены:
- общие понятия теории программ;
- языки представления знаний;
- языки обработки знаний.

Выделены следующие ключевые понятия и отношения: *язык программирования, технология программирования, программа, алгоритм, оператор, операнд, операция, переменная, константа, множество, кортеж, решаемый класс задач*, описание операции** и др.

Для представления синтаксиса и денотационной семантики базового языка программирования в базе знаний используется семантическая сеть. В виде семантической сети представлены:
- структуры программы базового языка программирования, ориентированного на обработку унифицированных семантических сетей
- структуры обрабатываемых информационных конструкций;
- структуры оператора базового языка программирования, ориентированного на обработку унифицированных семантических сетей;
- типов операторов генерации унифицированных семантических сетей;
- типов операторов удаления унифицированных семантических сетей;
- типов операторов ассоциативного поиска унифицированных семантических сетей;
- типов операторов проверки условий;
- типов операторов изменения свойств элементов унифицированных семантических сетей;
- типов операторов управления процессами обработки унифицированных семантических сетей;
- типов операторов вывода на консоль;
- типов операторов ассоциативной выборки;
- примеров программ базового языка программирования, ориентированного на обработку унифицированных семантических сетей.

Выделены следующие ключевые понятия и отношения: *scp-программа, scp-константа, scp-переменная, scp-параметр, трехэлементная sc-конструкция, пятиэлементная sc-конструкция, одноэлементная sc-конструкция, scp-оператор, тип scp-оператора, scp-оператор генерации sc-конструкций, scp-оператор ассоциативного поиска sc-конструкций, scp-оператор удаления sc-конструкций, scp-оператор проверки условия, scp-оператор изменения свойств, scp-оператор управления scp-процессом, scp-оператор ассоциативной выборки sc-конструкций, значение*, атрибутивное отношение assign_, атрибутивное отношение fixed_* и др. Также, в базу знаний включены утверждения и правила, описывающие способы проектирования программных текстов и задающие семантику языка.

Для каждого понятия выделены его свойства. Например, для понятия *scp-программа* выделены:
- синонимичные идентификаторы;
- пояснение;
- примеры (рисунок 1);
- структура программы.

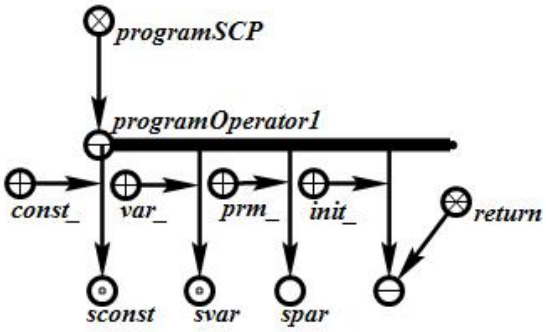

Рисунок 1 – SCP-программа с одним scp-оператором

Для понятия *тип scp-оператора* выделены

следующие свойства:

- синонимичные идентификаторы;
- определение;
- пояснение;
- разбиения по различным признакам (например, разбиение соответствующее типу выполняемого действия);
- структура (обобщенная для всех операторов);
- примеры.

Для понятия *genElStr3-оператор* выделены следующие свойства:

- синонимичные идентификаторы;
- пояснение;
- принадлежность;
- включение;
- компоненты;
- семантика;
- примеры с семантическими комментариями (рисунок 2).

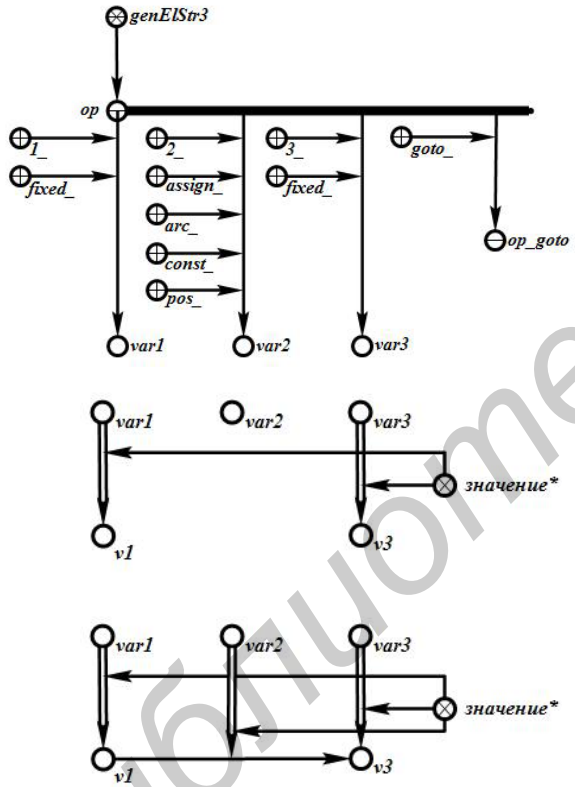

Рисунок 2 – Пример записи scp-оператора genElStr3, состояние памяти до и после выполнения оператора

При работе с системой пользователь может посмотреть синтаксис в виде семантической сети, в виде, представленном традиционно применяемыми средствами (форма Бэкуса-Наура и др.), в виде естественно-языковых текстов.

Для описания абстрактной машины, интерпретирующей базовый язык программирования, в базу знаний были включены понятия необходимые для представления в виде семантической сети:

- языковых средств, описывающих текущее состояние процесса интерпретации scp-программ;
- операций машины, отражающих специфику языка SCP (операционных семантик scp-операторов разных типов);
- используемых общих подпрограмм;
- модели sc-памяти;
- способов реализаций машины.

Выделены ключевые понятия: *scp-процесс, тип состояния scp-процесса, then-состояние scp-процесса, else-состояние scp-процесса, repeat-состояние scp-процесса, run-состояние scp-процесса, error-состояние scp-процесса, dead-состояние scp-процесса, sc-память, сегмент, операционная семантика genEl-оператора, операционная семантика genElStr3-оператора* и др. Для понятий выделены теоретико-множественные свойства. Например, для понятия *scp-процесс* определены синонимичные идентификаторы, пояснение, компоненты, примеры. Понятия для описания общих подпрограмм представляют собой имена программ, использующихся для реализации операционных семантик scp-операторов. Для каждой программы задана спецификация: пояснение, условие применения, класс задач, решаемых с помощью этой программы, алгоритм, примеры выполнения программы. Для описания понятий *операционная семантика* определенного scp-оператора выделены свойства: пояснение, алгоритм, примеры выполнения заданного оператора с семантическими комментариями.

Семантическая технология содержит множество типовых операций, которые можно использовать для реализации машины обработки знаний интеллектуальной help-системы по семантической теории программ. Множество операций машины обработки знаний help-системы было выделено на основании анализа спецификации вопросов. Каждому классу вопросов соответствует операция (типовая операция или специальная операция).

Заключение

Язык SCP в графодинамических ассоциативных машинах занимает особое место, поскольку его удобство и эффективность во многом определяют эффективность реализуемых с его помощью графодинамических моделей. Язык SCP является достаточно хорошо приспособленным к реализации преобразований, связанных с изоморфизмом sc-конструкций, благодаря ориентации языка SCP на переработку графовых конструкций, наличию в языке SCP операторов ассоциативного поиска, параллелизму.

Принципиальным отличием языка SCP от известных языков программирования является ориентация на графодинамическую (структурно перестраиваемую) ассоциативную память.

При разработке интеллектуальной help-системы для разработчиков программ, ориентированных на

обработку семантических систем, используется та же технология, что и использовалась для проектирования инструментальных средств. Это позволяет легко интегрировать систему с инструментальным средством и обеспечить информационное обслуживание разработчика scp-программ как при обучении, так и в процессе проектирования программного обеспечения. Язык программирования и язык представления знаний разработаны на базе той же технологии. Это позволяет унифицировать и интегрировать все необходимые знания о семантической теории программ.

Результаты, приведенные в работе, апробируются в рамках открытого проекта OSTIS (http://ostis.net).

Работа поддержана грантами БРФФИ-РРФИИ Ф10Р-149, Ф10Р-148, Ф10Р-175

Библиографический список

[Голенков В.В. и др., 2001a] Голенков, В.В. Представление и обработка знаний в графодинамических ассоциативных машинах. Монография / В.В. Голенков, О.Е. Елисеева, В.П. Ивашенко и др. Под ред. В.В. Голенкова. – Мн.: БГУИР, 2001. – 412 с.

[Электронный ресурс] Открытая семантическая технология компонентного проектирования интеллектуальных систем [Электронный ресурс]. – Электронные данные. – Режим доступа: http://ostis.net/mediawiki/index.php/.

[Голенков В.В. и др., 2001b] Программирование в ассоциативных машинах / В.В.Голенков, Г.С.Осипов, Н.А.Гулякина и др. – Мн.: БГУИР, 2001. – 276 с.

[Поспелов Д.А., 1990] Искусственный интеллект: В 3 кн. Кн. 2. Модели и методы: Справочник /Под ред. Д.А. Поспелова. - М.: Радио и связь, 1990.

[Гладун В.П., 1987] Гладун В.П. Планирование решений. - Киев: Наукова думка, 1987.

[Борщев В.Б., 1990] Борщев В.Б. Параллельные асинхронные вычисления и моделирование параллельного выполнения логических программ (вегетативная модель вычислений и её применения) // Итоги науки и техники. Сер. Вычислительные науки. Том 4.- М.: ВИНИТИ, 1990. - с. 114-197.

[Степанов А.М., 1981] Степанов А.М. Фреймы и параллельные смешанные вычисления. - Новосибирск, 1981. (Препринт / СО ВЦ АН СССР: N 297).

[Вагин В.Н., 1989] Вагин В.Н. Дедукция и обобщение в системах принятия решений. - М.: Наука, 1989.

[Кузнецов В.Е., 1989] Кузнецов В.Е. Представление в ЭВМ неформальных процедур. М.: Наука, 1989.

[Лазуркин Д.А. и др., 2010] Лазуркин Д.А., Пивоварчик О.В. Средства разработки программ, ориентированных на обработку семантических сетей // Информационные системы и технологии (IST"2010): материалы YI междунар. конф. (Минск, 24-25 нояб. 2010.- Минск: А.Н. Вараксин, 2010. - с. 501-504.

[Голенков В.В. и др., 1994a] Голенков В.В., Гулякина Н.А., Королев В.Г., Татаренко В.А., Золотой С.А.. Описание семантики языка SCPas. Операторы преобразования состояния SC-графа (Материалы по математическому обеспечению ЭВМ). – Мн: Ин-т техн. кибернетики АН Беларуси, 1994.

[Голенков В.В. и др., 1994b]Голенков В.В., Гулякина Н.А., Татаренко В.А., Гапонов П.А., Кузьмицкий В.М.. Описание семантики языка SCPas. Операторы поиска и проверки условий. Операторы управления вычислительным процессом (Материалы по математическому обеспечению ЭВМ). – Мн: Ин-т техн. кибернетики АН Беларуси, 1994.

LANGUAGES And TECHNOLOGIES PROGRAMMING, ORIENTED TO TREATMENT of SEMANTIC NETWORKS

Guliakina N.A., PivovarchykO.V., Lazurkin D.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

guliakina@bsuir.by
pivovarchyk@tut.by
lazurkin@bsuir.by

The article is devoted to the intellectual help-system for programmes developers which are oriented on handling semantic net. Semantic programming languages and programming technologies are described. The components of help-system are presented in the article. The intellectual help-system of semantic programming theory is described in details.

Introduction

Programmers use programming technologies which are oriented on handling knowledge base for engineering intellectual information systems. The use of this technologies requires special knowledge and a high level qualification. The use of intellectual help-systems will permit to decrease the basic requirements to the programmer's qualification and to increase the quantity of the engineered intellectual information systems.

Main Part

The article is devoted to the programming technology which are oriented on handling semantic nets and to the intellectual help-system. Sc-code is a knowledge representation language. SCP is a programming language.

The intellectual help-system is composed of components. Components contain formal models of technology's areas: a semantic programming theory, programmes' libraries, compute-aided software engineering, a methodology of software engineering, a training methodology of software engineering.

Now we engineer intellectual help-system of semantic programming theory. For help-system's development we use the open semantic technology for intelligent systems (OSTIS).

Conclusion

The knowledge representation language, the programming language, the compute-aided software engineering, the intellectual help-system develop using one technology. It will permit to integrate the help-system and the compute-aided software engineering, to make sure information service of scp-programmes' developer in the time of learning and in the time of engineering