

УДК 004.89

М.А. Князева, В.А. Тимченко

Институт автоматики и процессов управления ДВО РАН, г. Владивосток, Россия
mak@imcs.dvgu.ru, rakot2k@mail.primorye.ru.

Структурные редакторы программ на языках программирования высокого уровня и генератор моделей структурных программ в Банке знаний о преобразованиях программ*

Для решения образовательных, научных и практических проблем в области преобразований программ разработана концепция Специализированного банка знаний о преобразованиях программ в рамках нового научного направления «Компьютерные банки знаний». В статье описан современный взгляд на существующие структурные редакторы программ на языках программирования, после чего предложен подход к построению структурных редакторов программ на языках программирования и генератора программ на языке моделей структурных программ в рамках Специализированного банка.

Введение

Для решения научных, практических и образовательных проблем в области преобразования программ была предложена концепция Специализированного банка знаний о преобразованиях программ (СБкЗ_ПП) [1], основанная на результатах работ [2], [3] (рис. 1).

Общей задачей СБкЗ_ПП является централизация знаний о преобразованиях программ, координация процесса их обработки и коллективное развитие с целью получения наиболее качественных и современных знаний, а также облегчения их использования в науке, образовании и профессиональной деятельности.

Банк знаний о преобразованиях программ состоит из информационного наполнения (ИН) и оболочки; программного наполнения и блока администрирования. В состав информационного наполнения входят: онтология знаний о преобразованиях программ, онтологии языков программирования, онтология модели структурных программ, знания о преобразованиях программ, программы, представленные в терминах онтологий языков программирования, и программы, представленные в терминах онтологии МСП. Оболочка информационного наполнения предоставляет компьютерным программам доступ ко всей хранимой информации.

Чтобы формировать и развивать ИН, требуются средства редактирования. Такими средствами являются редакторы программ в терминах онтологий языков программирования, а также генератор моделей структурных программ (МСП) [4], [5], осуществляющий преобразования программ, представленных в терминах онтологии языка программирования в МСП. В работе [6] предложена концепция универсального редактора информации различных уровней общности (редактор ИРУО).

*Работа выполнена при финансовом содействии программы № 16 Президиума РАН, проект «Теоретические основы интеллектуальных систем, основанных на онтологиях, для интеллектуальной поддержки научных исследований», программы № 16 ОЭММПУ РАН, проект «Синтез интеллектуальных систем управления базами знаний и базами данных», и программы Дальневосточного отделения РАН (код проекта 05-III-Ж-01-008).

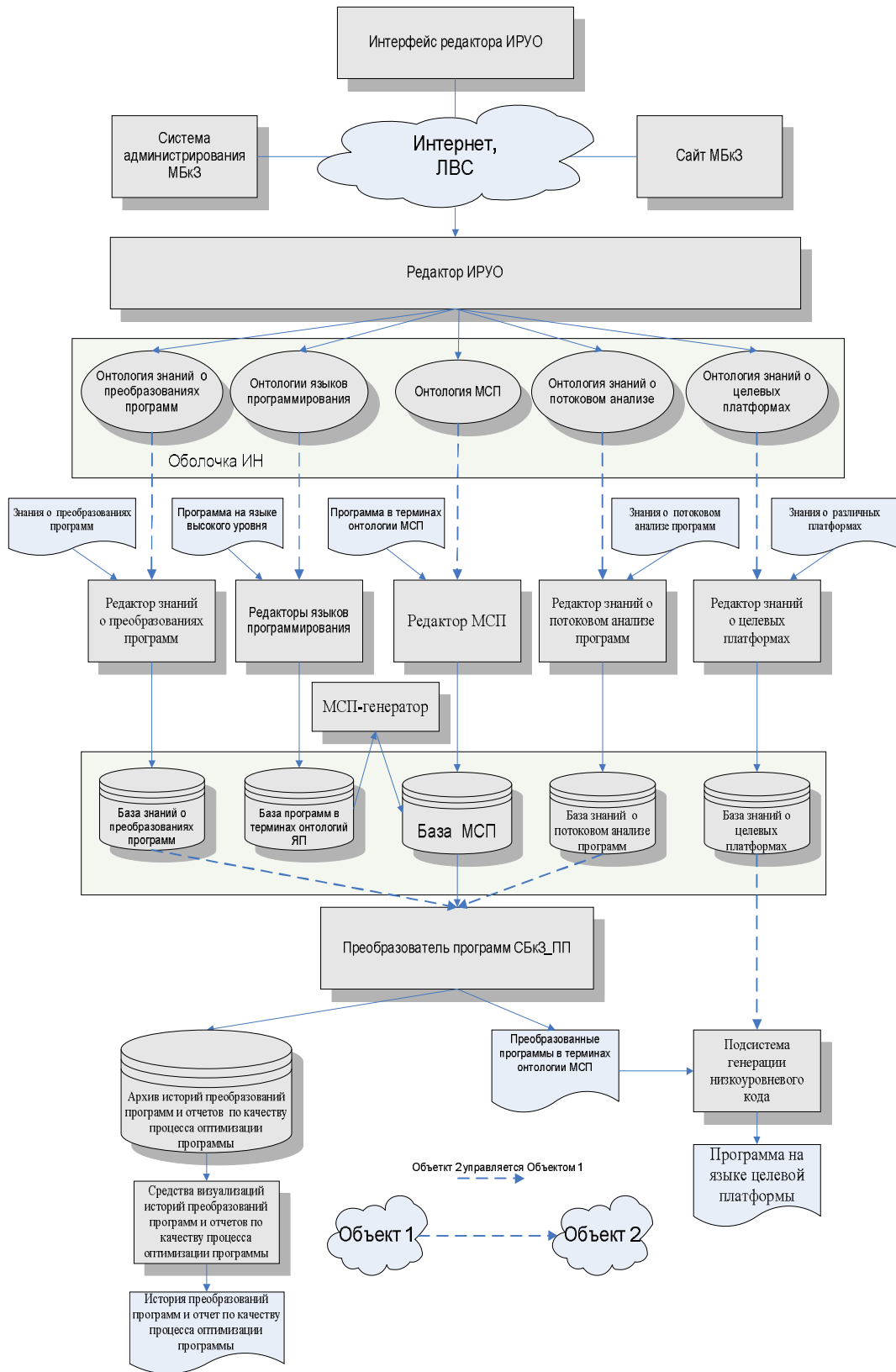


Рисунок 1 – Концепция СБкЗ_ПП

В статье предложен подход к созданию специализированных структурных редакторов программ на языках высокого уровня с использованием редактора ИРУО в качестве инструментального средства и генератора моделей структурных программ (МСП-генератора).

Анализ существующих средств структурного редактирования программ

В соответствии с концепцией Банка знаний о преобразованиях программ редактором программ должен быть структурный редактор. Название «структурный» связано с тем, что такие редакторы работают с логической структурой программы, задаваемой обычно графом специального вида [7-16], а текст программы или ее фрагментов строится лишь по мере необходимости. Структурные редакторы позволяют применять технологию пошаговой детализации программы в явном виде, т.е. предоставляют для этого специальные средства. При работе со структурным редактором пользователь освобождается от детального знания синтаксиса используемого языка программирования, а также обеспечивается определенная степень правильности редактируемой программы (от правильности общей структуры до полной синтаксической и семантической корректности).

Существующие структурные редакторы обладают следующими функциональными возможностями:

- поддерживают режим редактирования программы в текстовом представлении (только самые первые структурные редакторы не предоставляли такой возможности);
- преобразование текстового (или другого визуализируемого) представления программы в ее представление в виде дерева абстрактного синтаксиса;
- обратное преобразование программы – из дерева абстрактного синтаксиса в визуализируемое (чаще всего, текстовое);
- непрерывный (полный) синтаксический и частичный семантический контроль в процессе создания и редактирования программы;
- позволяют сохранять тексты программ (или их фрагменты) на внешних носителях как в обычном текстовом файле, так и в некотором специфичном (для конкретного структурного редактора) формате, содержащем информацию о синтаксической структуре текста.

Появившиеся относительно недавно структурные редакторы, кроме всего прочего:

- не зависят от набора поддерживаемых языков и способны динамически загружать новый язык в диалоге с пользователем;
- поддерживают метаязык для описания синтаксиса языков программирования и настройки режимов редактора на описываемый язык.

Архитектура большинства существующих структурных редакторов укладывается в следующую схему [8-13]. Редактор состоит из компонент: подсистема диалога с пользователем (монитор), она же является главной управляющей программой редактора; подсистема работы с архивом (например, файлами); подсистема распознавания (разбора) визуализируемого представления программы (как правило, текстового); подсистема генерации визуализации (преобразования дерева), которая осуществляет преобразование древовидного представления программы в визуализи-

руемое, а также подсистемы перемещения по дереву и манипуляций с деревом (имеется в виду дерево абстрактного синтаксиса программы).

Основной принцип структурного редактирования заключается в том, что строится и модифицируется не текст программы, а ее дерево абстрактного синтаксиса (ДАС) [7-10], [14], [15]. Редактирование, таким образом, сводится к манипуляциям с поддеревьями ДАС программы, которые могут быть получены как структурным, так и текстовым способом. При первом способе пользователь выбирает требуемую конструкцию или ее заготовку (шаблон) из набора возможных (или допустимых) конструкций, при втором он вводит ее в текстовом виде.

Дерево абстрактного синтаксиса, с которым работает структурный редактор, можно использовать для немедленного выполнения программы в режиме интерпретации или для генерации объектного кода. С этой точки зрения структурный редактор берет на себя функции синтаксического блока компилятора или интерпретатора.

Из существующих инструментальных систем поддержки разработки структурных редакторов можно отметить системы SG, КОРСАР [17], которые были использованы при разработке ряда структурных редакторов для таких языков программирования, как Паскаль, Эль-76, Фортран, Оберон-2, а также систему ALOE, которая является составной частью более широкого проекта GANDALF.

Подводя итоги, необходимо отметить, что анализ существующих редакторов показал имеющиеся в них следующие недостатки: разные редакторы создают разные представления ДАС в зависимости от используемого в них языка программирования, они не удовлетворяют современным требованиям к пользовательскому интерфейсу и не приспособлены для работы в среде Интернет.

Разработанные для Банка знаний о преобразованиях программ специализированные редакторы обладают развитым графическим пользовательским интерфейсом и могут функционировать в среде Интернет.

Чтобы получить представление программы, не зависящее от используемых редакторов и языков программирования, необходимо программное средство для формирования такого представления исходной программы.

Таким программным средством является МСП-генератор – программа, которая по исходному представлению ДАС программы в терминах онтологии языка программирования высокого уровня, согласно заложенной в нее схеме (правилам) преобразования, генерирует ДАС МСП.

Специализированные редакторы программ в терминах онтологий языков программирования

Для создания специализированных структурных редакторов программ на языках программирования высокого уровня средствами Редактора ИРУО необходимы модели онтологий этих языков программирования. Такие модели были разработаны и представлены в терминах языка ИРУО.

На рис. 2 представлен фрагмент модели онтологии языка С.

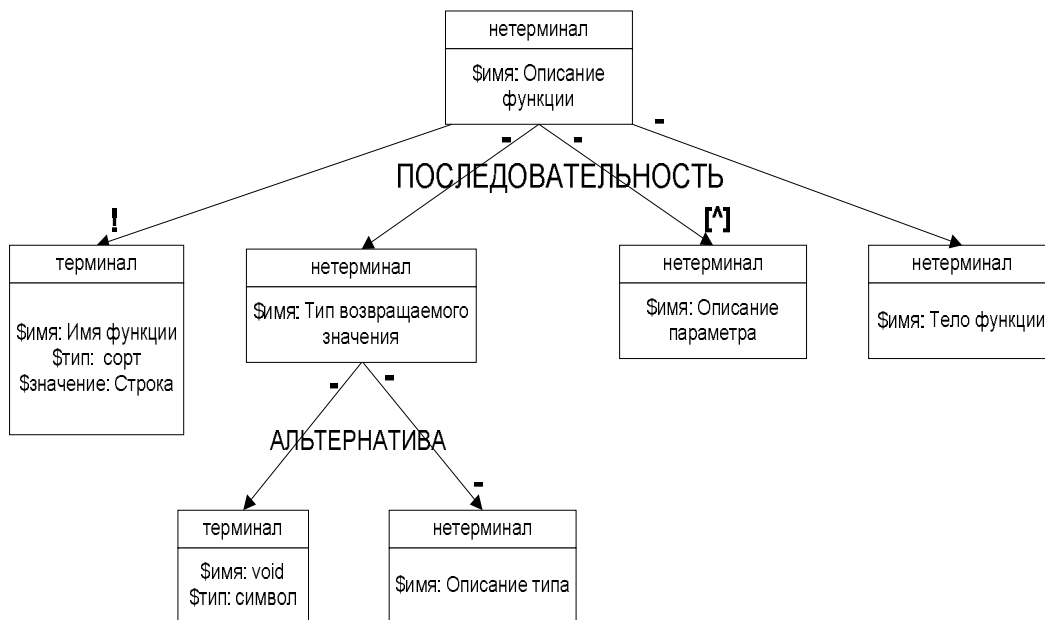


Рисунок 2 – Фрагмент модели онтологии языка С («Описание функции»)

Эта модель представляет собой семантическую сеть понятий – набор понятий и направленных отношений между ними, снабженных специальной разметкой. Понятия (или вершины) сети представляют термины онтологии С, а направленные дуги описывают, каким образом одни термины онтологии определены через другие. Термин разметки «ПОСЛЕДОВАТЕЛЬНОСТЬ» говорит о том, что для описания предка должны быть описаны все потомки, в то время как термин «АЛЬТЕРНАТИВА» говорит о том, что для описания предка достаточно описать одного из потомков.

На рис. 3 представлен фрагмент модели онтологии языка Паскаль.

Разработанные модели онтологий языков программирования с помощью редактора ИРУО заносятся в информационное наполнение банка знаний, и на их основе создаются специализированные редакторы программ на языках С и Паскаль. Модели используются в качестве входной информации (метаинформации) для редактора ИРУО: они являются своего рода программами, а редактор ИРУО – интерпретатором этих программ. Так, в процессе интерпретации модели онтологии языка Паскаль редактор ИРУО становится по существу специализированным структурным редактором программ на языке Паскаль.

Генератор программ в терминах онтологии модели структурных программ

В ходе разработки МСП-генератора был разработан алгоритм генерации МСП по ДАС программы в терминах онтологии языка программирования высокого уровня. В основу алгоритма положена схема преобразования фрагментов ДАС программы в терминах онтологии языка программирования высокого уровня во множество фрагментов ДАС МСП. Алгоритм осуществляет обход ДАС программы в терминах онтологии языка программирования высокого уровня (сформированное специализированным редактором программ) в ширину и обрабатывает каждую вершину этого дерева согласно схеме преобразования фрагментов ДАС программы в терминах онтологии языка программирования высокого уровня во множество фрагментов ДАС МСП.

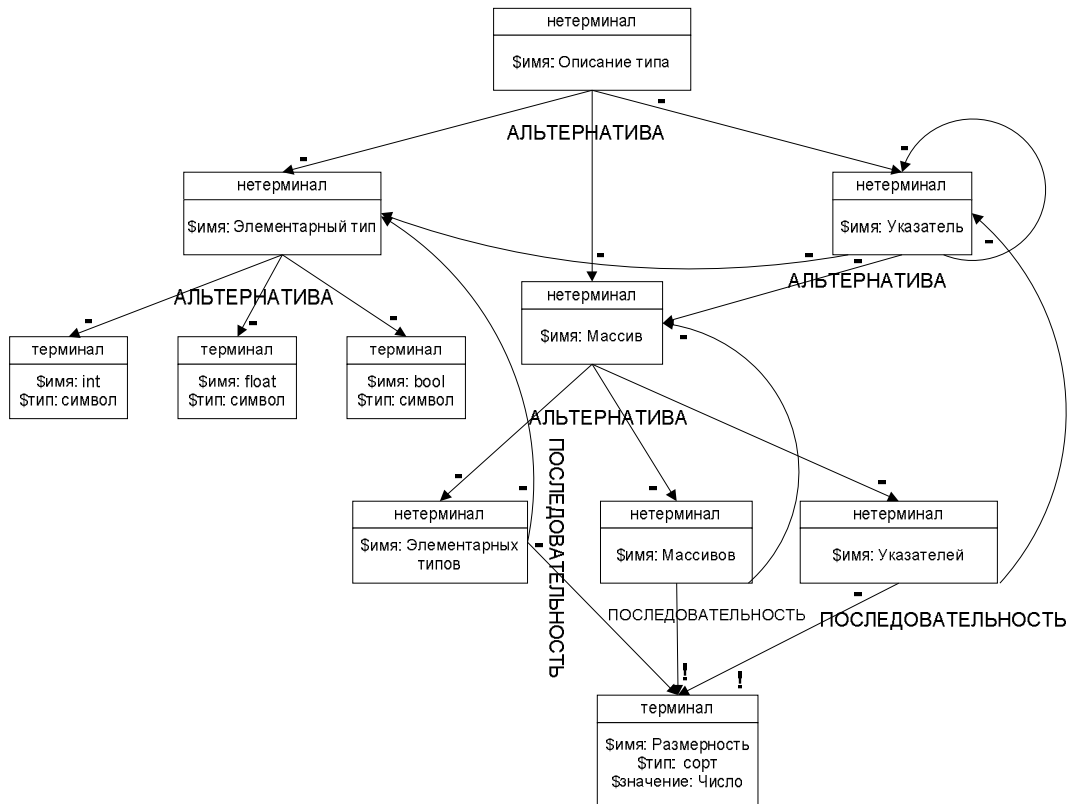


Рисунок 3 – Фрагмент модели онтологии языка Паскаль («Описание типа»)

Пример преобразования фрагмента «Программа» – это термин онтологии языка Паскаль – представлен ниже.

МСП (Программа) = Фрагмент программы в терминах МСП. Графическое изображение фрагмента представлено на рис. 4.

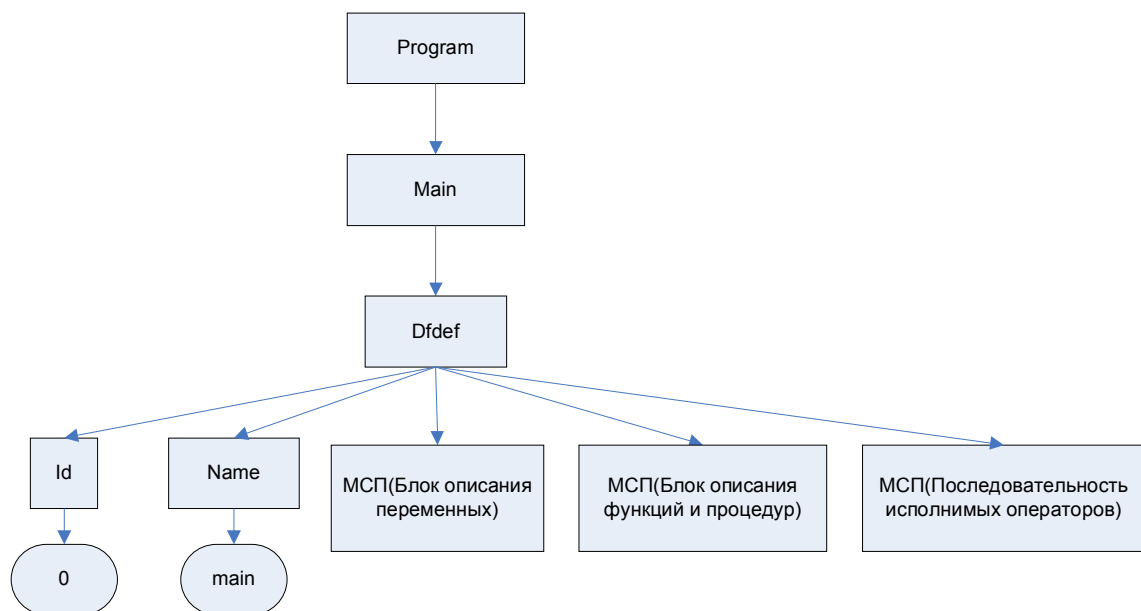


Рисунок 4 – Преобразование фрагмента «Программа»

МСП – это отображение: фрагмент программы в терминах онтологии Паскаль → фрагмент программы в терминах МСП.

Областью определения данного отображения является множество фрагментов программы (фрагментом является и сама программа) в терминах онтологии языка программирования Паскаль. Областью значения данного отображения является множество фрагментов программы на языке моделей структурных программ.

Пример преобразования фрагмента «Оператор присваивания» – это термин онтологии языка С – представлен ниже.

МСП (Оператор присваивания) = Фрагмент программы в терминах МСП. Графическое изображение фрагмента представлено на рис. 5.

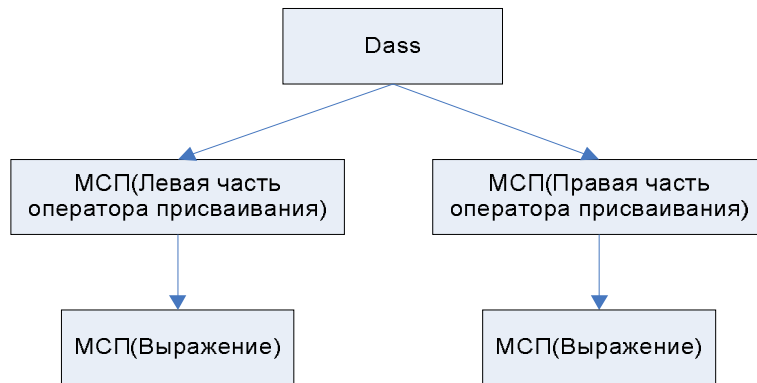


Рисунок 5 – Преобразование фрагмента «Оператор присваивания»

МСП – это отображение: фрагмент программы в терминах онтологии С → фрагмент программы в терминах МСП.

Областью определения данного отображения является множество фрагментов программы в терминах онтологии языка программирования С. Областью значения данного отображения является множество фрагментов программы на языке моделей структурных программ.

МСП-генератор реализован на языке Java, отлажен в комплексе с другими компонентами Специализированного банка и оттестирован на ДАС программ в терминах онтологии языка Паскаль, которые были сформированы соответствующим редактором программ.

После создания МСП-генератора были проведены экспериментальные исследования его рабочих характеристик. Далее описана среда проведения экспериментов.

На стороне клиента: Intel Pentium4 2.8 ГГц, 512 Мб ОЗУ. ОС MS Windows XP SP2.

На стороне сервера: Intel Pentium4 2.8 ГГц, 1024 Мб ОЗУ. ОС MS Windows 2000 Server SP4.

На рис. 6 показана схема применения компонентов СбкЗ_ПП, которая используется для проведения экспериментов (цифрами обозначен порядок вызовов компонентов).

Результаты измерений для трех ДАС программ в терминах онтологии языка Паскаль представлены на рис. 7.

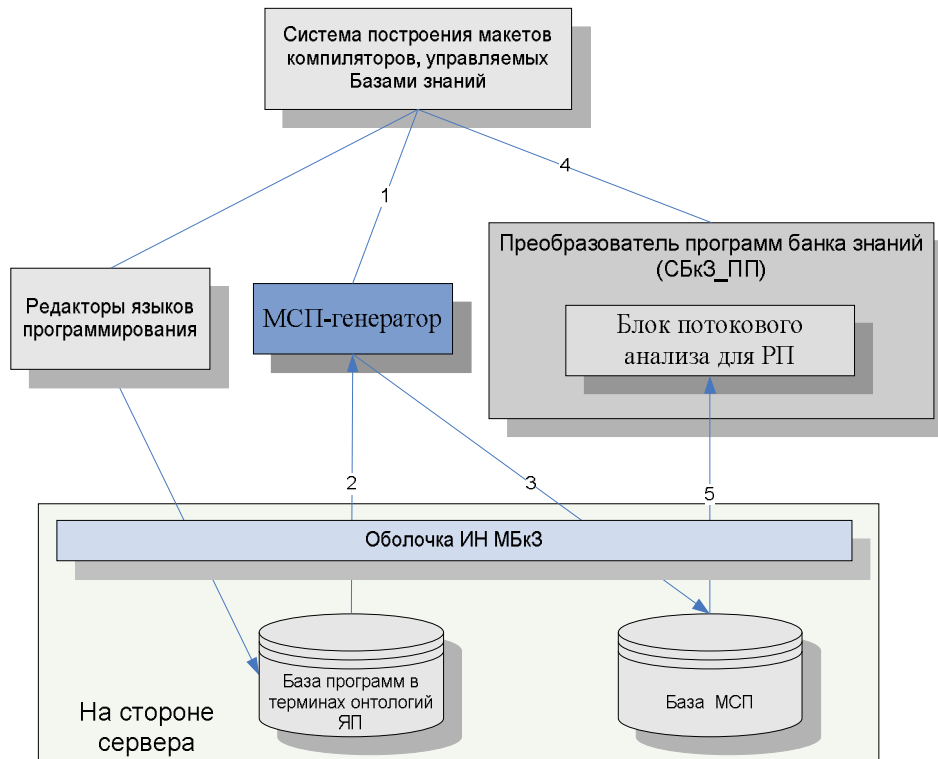


Рисунок 6 – Схема применения компонентов СБкЗ_ПП в экспериментах

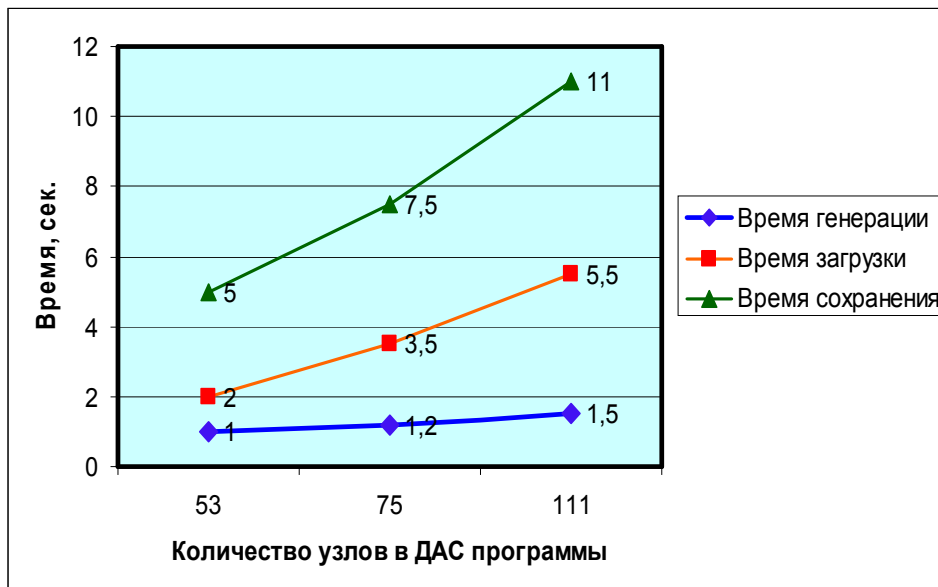


Рисунок 7 – Результаты экспериментов

Измерялись время загрузки программ в терминах онтологии языка Паскаль из Базы программ в терминах онтологий языков программирования, время генерации ДАС МСП и время записи сгенерированных ДАС МСП в базу МСП.

Заключение

В работе на основе представленных результатов анализа существующих структурных редакторов программ предложен подход к созданию специализированных

редакторов программ на языках программирования высокого уровня, обладающих практически всеми преимуществами существующих структурных редакторов и лишенных выявленных при анализе недостатков. Экспериментально исследованы возможность применения и эффективность предложенного подхода.

Литература

1. Князева М.А., Клещев А.С. Управление информацией о преобразованиях программ. Часть 1: Анализ проблем и пути их решения на основе методов искусственного интеллекта // Изв. РАН. ТИСУ. – 2005. – № 5.
2. Орлов В.А., Клещев А.С. Многоцелевой банк знаний. Часть 1: Концепция и политика. – Владивосток: ИАПУ ДВО РАН, 2003. – 40 с. (доступна по адресу <http://www.iacp.dvo.ru/es/>).
3. Орлов В.А., Клещев А.С. Многоцелевой банк знаний. Часть 2: Требования. – Владивосток: ИАПУ ДВО РАН, 2003. – 40 с. (доступна по адресу <http://www.iacp.dvo.ru/es/>).
4. Артемьева И.Л., Князева М.А., Купневич О.А. Модель онтологии предметной области «Оптимизация последовательных программ». Ч. 1: Термины для описания объекта оптимизации // НТИ. Сер. 2. – 2002. – № 12 – С. 23-28.
5. Артемьева И.Л., Князева М.А., Купневич О.А. Модель онтологии предметной области «Оптимизация последовательных программ». Ч. 2: Термины для описания процесса оптимизации // НТИ. Сер. 2. – 2003. – № 1. – С. 22-29.
6. Орлов В.А., Клещев А.С. Многоцелевой банк знаний. Часть 3. Концепция универсального Редактора ИРЮ. – Владивосток: ИАПУ ДВО РАН, 2003. – 28 с. (доступна по адресу <http://www.iacp.dvo.ru/es/>).
7. Жоголев Е.А. Синтаксически управляемое конструирование программ // Программирование. – 1979. – № 6.
8. Мучник Т.Г. Языково-настраиваемый структурный редактор со средствами семантического контроля // Программирование. – 1990. – № 2.
9. Александров С.Ю., Скопин И.Н. Синтаксически-ориентированный редактор SED: Препр. № 25 / ТМ и ВТ АН СССР. – Новосибирск: 1989. – 30 с.
10. Александров С.Ю. Синтаксически-ориентированные редакторы: функциональные возможности и архитектура: Препр. № 3 / ИТМ и ВТ АН СССР. – Новосибирск: 1987. – 35 с.
11. Кузьминов Т.В. Языково-настраиваемый структурный редактор для ЭВМ КРОНОС: Препр. № 1 / Институт систем информатики АН СССР. – Новосибирск: 1990. – 29 с.
12. Александров С.Ю., Скопин И.Н. Архитектура синтаксически-ориентированного редактора // Теория и практика систем информатики и программирования / Под ред. А.П. Ершова. – Новосибирск: НГУ, 1988.
13. Анисков М.И. Реализация языково-ориентированного редактора программ // Среда программирования: методы и инструменты / Под ред. И.В. Поттосина. – Новосибирск: Ин-т систем информатики СО РАН, 1992.
14. Болтаев Т.Б., Кузьминов Т.В., Поттосин И.В. О структурном конструировании и инструментах его поддержки // Среда программирования: методы и инструменты / Под ред. И.В. Поттосина. – Новосибирск: Институт систем информатики СО РАН, 1992.
15. Ильин М.А., Самочадин А.В., Цикин С.И. Структурный редактор, ориентированный на сборочно-конкретизирующее программирование // Методы трансляции и конструирования программ / Под ред. А.П. Ершова. – Новосибирск: Вычислительный центр СО АН СССР, 1988.
16. Кочетов Д.В. Реализация языковой подсистемы в структурном редакторе // Инструменты и методы разработки программ / Под ред. И.В. Поттосина. – Новосибирск: Ин-т систем автоматки СО РАН, 1994.
17. Александров С.Ю. Инструментальная поддержка конструирования синтаксически-ориентированных редакторов // Методы трансляции и конструирования программ / Под ред. А.П. Ершова. – Новосибирск: Вычислительный центр СО АН СССР, 1988.

М.А. Knyazeva, V.A. Timchenko

Structural Programs Editors in the High-level Programming Languages and Generator of Structural Programs' Models within the Knowledge Bank of Programs' re-organization

The concept of Specialized knowledge bank of program transformations is developed in order to solve educational, scientific and practical problems in the field of program transformations according to a new scientific direction «Computer knowledge banks». The modern point of view on existing structured program editors in programming languages is presented in the paper. Then the approach to construction of structured programs, editors in programming languages and the generator of programs in model of structural programs language within the framework of the Specialized bank is suggested.

Статья поступила в редакцию 14.07.2005.