

РАЗРАБОТКА КОНВЕРТЕРА КОДОВ ВЫЧИСЛИТЕЛЬНЫХ МОДУЛЕЙ С ЯЗЫКА PL/1 НА ЯЗЫК С#

А.В. Мартинкевич

Самарский государственный аэрокосмический университет им. С.П. Королева

Введение

В настоящее время существует огромное количество программ, написанных на языках программирования, которые считаются устаревшими.

К ним относится программное обеспечение систем автоматизированного проектирования (САПР) летательных аппаратов и двигателей, разработанное в 70-90 годах XX века СГАУ с использованием ЭВМ серии ЕС. Фактически использование хорошо отлаженных и абсолютно правильно работающих программ в настоящее время зачастую невозможно, но даже при наличии такой возможности сопровождение и развитие подобных программ является трудоемкой задачей, требующей привлечения высококвалифицированных специалистов, имеющих знания малоиспользуемого в текущее время языка.

Для решения этой проблемы необходимо создание программного комплекса, который правильно, удобно и быстро позволял бы вводить, конвертировать и тестировать исходную программу, преобразованную с языка PL/1 в язык С#. Эти задачи часто кажутся простыми и недооцениваются. Большая часть проектов по преобразованию программ на современные языки программирования является убыточной. Полностью удовлетворительный результат достигается крайне редко. Последнее возникает из-за неправильной оценки сложности преобразования программ.

Проблемы языковых преобразований

Обучение программированию ориентировано исключительно на новые разработки, а не на улучшение существующих, не говоря уже о сопровождении унаследованных приложений. Эта проблема настолько серьезна, что Каперс Джонс упоминает ее как один из 60 главных рисков процесса разработки программного обеспечения [1]. Языковой конвертер решает проблемы с персоналом: преобразование языков ликвидирует разрыв между знаниями обычного программиста и знаниями, необходимыми для решения проблем унаследованных систем.

Задача конвертирования выглядит достаточно простой, на первой взгляд, и зачастую оценка затрат, необходимых на разработку конвертера является ошибочной. Сейчас многие организации предлагают свои услуги по конвертированию программ на различных языках, но большинство из них не является специалистами в этом деле. Часто конвертеры либо работают абсолютно правильно лишь в простейших ситуациях, либо не работают вовсе. Некоторые организации заявляют, что занимаются автоматической конвертацией программ, даже не обладая необходимыми программными средствами.

Существует огромное количество примеров, когда компании, взявшиеся за написание преобразователя из одного языка в другой, не довели дело до конца [2]. Большая часть проектов подобного рода является экономически провальными. Ошибки в оценке сложности задачи преобразования иногда достигают невероятных размеров.

С. Спронг, работавший над переносом программ из Фортрана в Си, высказал в дискуссионной группе alt.folklore.computers [3] следующее мнение: «Низкоуровневый перенос программ, даже при наличии документации, является одной из самых темных разновидностей программистского колдовства; знание о том, как это правильно делать, гарантирует вам прямое попадание в ад».

На первый взгляд, автоматическое изменение структуры программы, связанное с изменением парадигмы программирования, теоретически возможно. Однако этот процесс очень трудно автоматизировать, в нем подразумевается активное участие человека [2]. Вместе с тем польза от таких вложений значительно превышает затраты, особенно для систем с длительным временем жизни.

Алгоритм конвертации кодов вычислительных модулей с языка PL/1 на язык C#

Наиболее общая схема алгоритма конвертации кодов вычислительных модулей с языка PL/1 на язык C#, выполненная в системе визуального проектирования GRAPH, представлена на рис. 1.

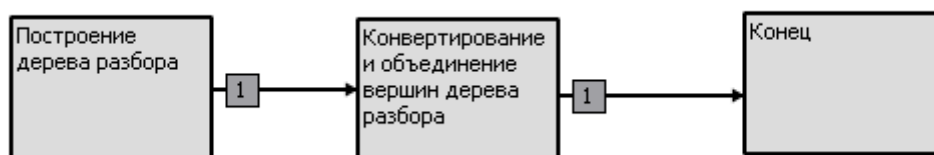


Рис. 1. Наиболее общая схема алгоритма конвертации кодов вычислительных модулей с языка PL/1 на язык C#

На рис. 2 изображена общая схема последовательного алгоритма построения дерева разбора.

В языке PL/1 наиболее синтаксически независимыми друг от друга элементами являются процедуры, что позволяет легко распараллелить конвертацию отдельных процедур. На 2-м этапе построения дерева разбора происходит выделение всех процедур программы. Далее все они начинают обрабатываться независимо и параллельно. Затем после построения дерева разбора для конкретной процедуры начинается процесс конвертации данной процедуры без необходимости ожидания завершения обработки других процедур.

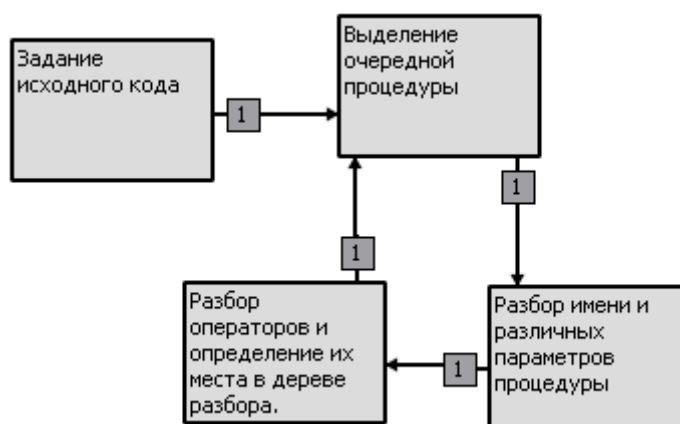


Рис. 2. Общая схема последовательного алгоритма построения дерева разбора

На рис. 3 представлена общая схема последовательного алгоритма конвертации и объединения вершин дерева разбора.

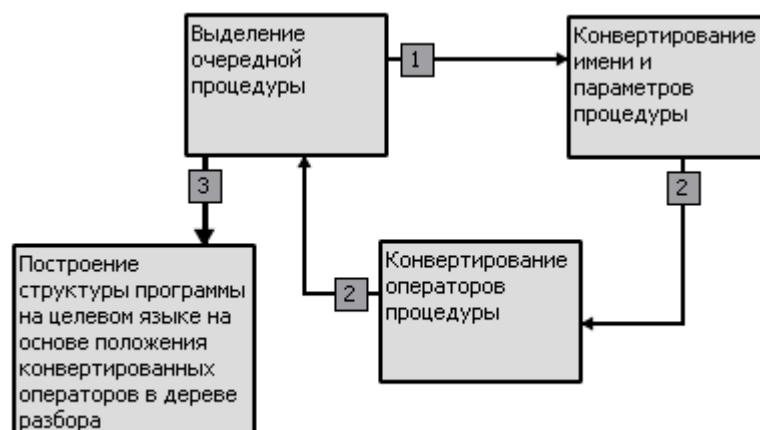


Рис. 3. Общая схема последовательного алгоритма конвертации и объединения вершин дерева разбора

Одним из основных этапов в процессе конвертации процедуры является конвертация отдельных операторов. Операторы могут быть конвертированы абсолютно независимо. Таким образом, все множество операторов исходной программы можно собрать в некоторое количество групп заданных размеров, что позволяет обрабатывать их отдельно на разных процессорах параллельно и независимо.

Структурно-функциональная схема

На рис. 4 изображена структурно-функциональная схема программного комплекса, который позволяет осуществлять конвертирование кодов вычислительных модулей с языка PL/1 на язык C#.

Программный комплекс состоит из следующих функциональных частей:

1. Система распознавания текста. В созданном программном комплексе используется интеллектуальная система оптического распознавания текста ABBYY FineReader, которая при помощи удобного интерфейса позволяет преобразовывать различные типы документов (например, бумажные документы, PDF-файлы или фотографии текстов, полученные при помощи цифровой фотокамеры) в редактируемый вид.
2. Система сравнения текстов. Специально для работы данного программного комплекса была написана система, которая позволяет построчно сравнивать любые два исходных текста и в удобной визуальной форме отображать различия между ними.
3. Система конвертирования программ с языка PL/1 на язык C#. В ходе создания программного комплекса был также создан конвертер, который позволяет конвертировать исходные коды, написанные на языке PL/1, в исходные коды, написанные на языке C#.
4. Система компилирования и исполнения исходных кодов на языке C#. Система позволяет проводить компиляцию программ, написанных на языке C#, непосредственно во время работы программного комплекса. Полученные в результате компиляции исполняемые файлы настраиваются и запускаются также при помощи данной системы.
5. Система аутентификации пользователей и хранения истории действий. Данная система позволяет ограничить доступ к программному комплексу, а также позволяет хранить полную историю действий различных пользователей. Поддерживает версию произведенных изменений.
6. Система компилирования и исполнения исходных кодов на языке C#. Система позволяет проводить компиляцию программ, написанных на языке C#, непосредственно во время работы программного комплекса.

но во время работы программного комплекса. Полученные в результате компиляции исполняемые файлы настраиваются и запускаются также при помощи данной системы.

7. Система аутентификации пользователей и хранения истории действий. Данная система позволяет ограничить доступ к программному комплексу, а также позволяет хранить полную историю действий различных пользователей. Поддерживает версию произведенных изменений.
8. Система управления реляционными базами данных. В качестве системы управления базами данных в программном комплексе используется Microsoft SQL Server.

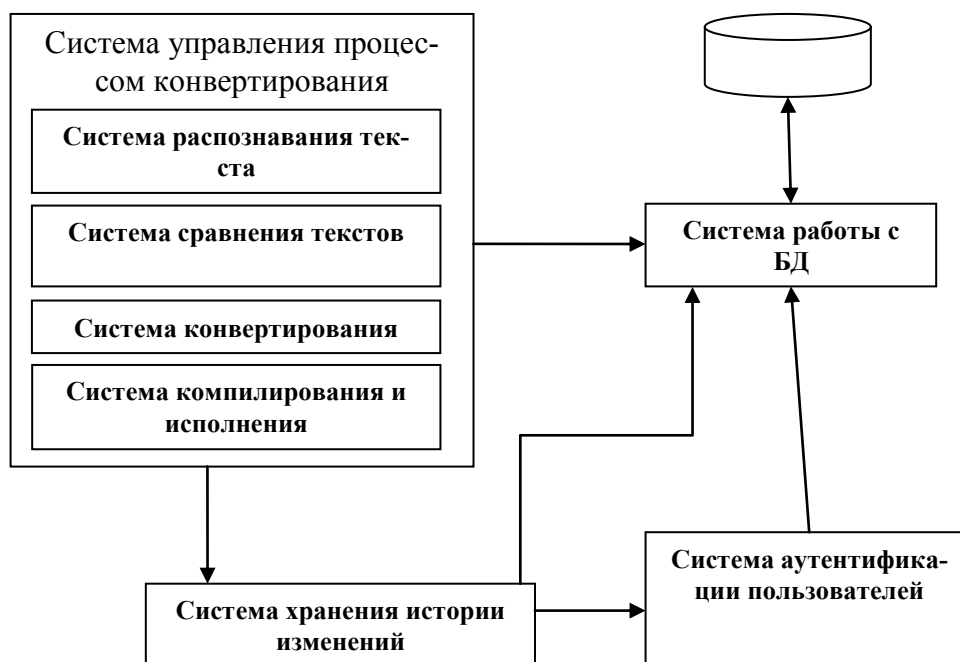


Рис. 4. Структурно-функциональная схема

Заключение

В докладе описывается конвертер кодов вычислительных модулей с языка PL\1 на язык C#, который обладает следующими свойствами:

1. Распознает код модуля на языке PL\1 с бумажного носителя.
2. Производит сравнения текстов полученных из двух разных источников и визуализирует результаты сравнения.
3. Строит дерево разбора кода на языке PL\1.
4. Конвертирует дерево разбора в код на языке C#.
5. Проверяет наличие синтаксических ошибок в конвертированной программе.
6. Компилирует конвертированную программу.
7. Позволяет проводить тестирование конвертированной программы.
8. Хранит историю изменений кодов в процессе конвертирования.
9. Обладает системой авторизации.

Литература

1. Jones C. Assessment and Control of Software Risks, Prentice Hall, Englewood Cliffs. N.J., 1994.

2. Kontogiannis K. et al. Code Migration through Transformations: An Experience Report // Proc. IBM Center for Advanced Studies Conf. (CASCON-98). IBM. Armonk, NY, 1998, P. 1-12.
3. Дискуссионная интернет-группа google –
<http://groups.google.com/group/alt.folklore.computers/topics>.