УДК 621.372

РАЗРАБОТКА МЕТАКОМПИЛЯТОРА RVTI-ГРАММАТИКИ НА ПРИМЕРЕ SHAREPOINT

Н.Н. Войт[15], В.А. Гордеев [16], В.С. Хородов, А.С. Степанов [17]

Предложен метакомпилятор диаграмматических языков на основе RVTI-грамматики. Метакомпилятор позволяет автономно строить грамматику диаграмматического языка и генерировать операции над внутренней памятью.

Введение

Использование метакомпилятора в графических грамматиках позволяет упростить работу с правилами грамматики и сохранить эффективность парсинга диаграмм.

Общая структура метакомпиляторов диаграмматических языков

Метакомпилятор часто называют компилятор компилятором.

Компилятор компиляторов – программа, принимающая на вход синтаксическое или семантическое описание языка программирования и на выходе формирующая компилятор для этого языка[1].

В данном случае на вход подаются правила описанные в форме Бэкуса-Наура (БНФ), так как эта форма позволяет покрыть большинство особенностей диаграмматических языков.

Метакомпилятор представляет собой приложение, написанное на C# с использованием GOLD Parser Builder [2] для разбора языка метаописаний. Структура взаимодействия его компонентов в составе метакомпилятора показана на рис. 1 [3]

[15] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: n.voit@ulstu.ru*

[16] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: gorde-vlad@yandex.ru*

[17] *432027, Ульяновск, ул. Северный Венец, 32, УлГТУ, e-mail: step_al_ul@mail.ru*

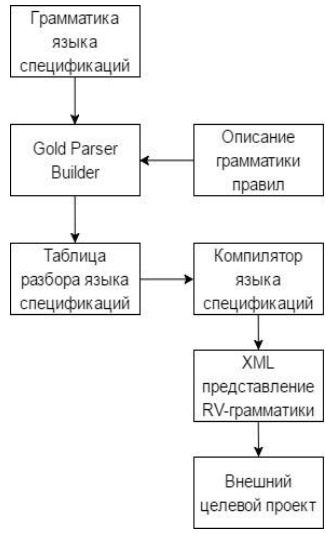


Рис.1. Структура метакомпилятора диаграмматических языков с использованием Gold Parser Builder

В метакомпиляцию входят следующие этапы:

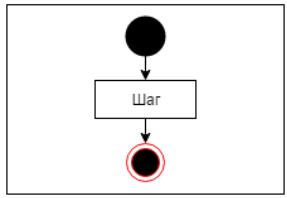
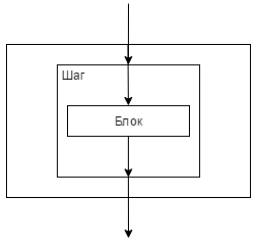
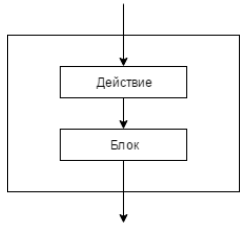
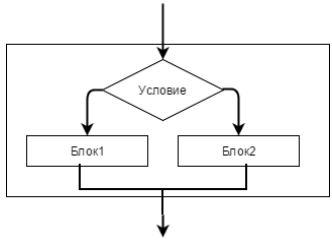
1. Лексический анализ
2. Синтаксический анализ и построение дерева разбора
3. Анализ дерева разбора и построение АСГ
4. Тансляция – преобразование АСГ в термины RV-грамматики
5. минимизация и оптимизация
6. RV-грамматика сохраняется в формате XML

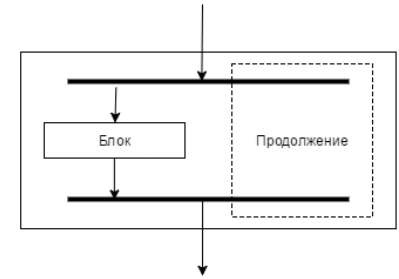
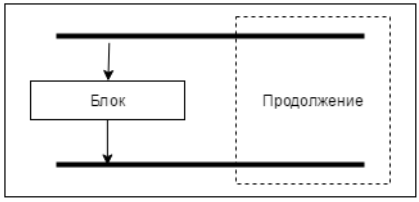
Правила описания синтаксиса языка диаграмм SharePoint

Перед тем как анализировать правила их необходимо составить. Данные правила были основаны на правилах расширяющих нотацию БНФ. Это расширение позволит привести описание графических конструкций к виду, наиболее похожему на их визуальное представление. В правой части правила идет разделение на две группы – блоки и связи между ними. Связи, так же, делятся на внешние, связывающие правило с «внешними» элементами и внутренние, связывающие элементы внутри.[4]

В таблице 1 представлены полученные конструкции бизнес-процесса SharePoint и их метаописания.

Таблица 1. Правила описания БНФ для языка SharePoint

Графическое представление правила	Текстовое описание правила
	**Rule:** IdDiagram **Consist of:** IdBegin ElBegin, IdSteps ElStep, IdEnd ElEnd **Internal Relationships:** ElStep.In = ElBegin.Out, ElStep.Out = ElEnd.In **External Relationships: ;**
	**Rule:** IdSteps **Consist of:** IdBeginStep ElBeginStep, IdBlock ElBlock, IdEndStep ElEndStep **Internal Relationships:** ElBlock.In = ElBeginStep.Out, ElBlock.Out = ElEndStep.In **External Relationships:** IdSteps.In = ElBeginStep.In, ElEndStep.Out = IdSteps.Out;
	**Rule:** IdBlock **Consist of:** IdAction ElAction, IdBlock ElBlock **Internal Relationships:** ElBlock.In = ElAction.Out **External Relationships:** ElBlock.In = ElAction.In, IdBlock.Out = ElBlock.Out
	**Rule:** IdBlock **Consist of:** IdBlock ElBlock1, IdBlock ElBlock2, IdIf ElIf, IdJoin ElJoin **Internal Relationships:** ElIf.In1 = ElBlock1.In, ElIf.Out2 = ElBlock2.In, ElJoin.In1 = ElBlock1.In, ElJoin.In2 = ElBlock2.In **External Relationships:** IdBlock.In = ElIf.In, IdBlock.Out = ElJoin.Out

Графическое представление правила	Текстовое описание правила
	**Rule:** IdBlock **Consist of:** IdBlock ElBlock, IdParal ElParal, IdJoinParal ElJoinParal, IdCont ElCont **Internal Relationships:** ElParal.Out1 = ElBlock.In, ElParal = ElCont. In, ElJoinParal = ElCont. Out **External Relationships:** ElBlock.In = ElParal.In, ElBlock.Out = ElJoinParal.Out
	**Rule:** IdCont **Consist of:** IdBlock ElBlock, IdParal ElParal, IdJoinParal ElJoinParal, IdCont ElCont **Internal Relationships:** ElParal.Out1 = ElBlock.In, ElParal = ElCont.In, ElJoinParal = ElCont.Out **External Relationships:** ElCont.In = ElParal, ElCont.Out = IdJoinParal

Основным правилом является первое, оно описывает диаграмматический язык в целом. Далее правила описывают множество допустимых блоков в диаграмме SharePoint. Правило «Шаги» описывает особенность языка, что все блоки должны быть внутри «Шаг». Последнее правило «Продолжение» рекурсивно ссылается на себя, благодаря чему может содержать несколько параллельных ветвей «Блоки».

Правила описания синтаксиса языка диаграмм SharePoint

Алфавит Квазитермов получается из части правила, начинающегося с Consist of. Алфавит строится в два прохода: построение множества имен и фильтрация множества имен. При первом проходе собираются все имена из правил. Чтобы этого достичь из каждого правила в части Consist of собираются множество используемых имен. В процессе прохода строится дополнительное множество имен правил. Пример разбора правил приведен в Таблица 2.

Таблица 2. Формирование квазитермального алфазита

Правило	Множество имен	Множество имен правил
Rule: IdDiagram **Consist of:** IdBegin ElBegin, IdSteps ElStep, IdEnd ElEnd **Internal Relationships:**	ElBegin, ElStep, ElEnd	IdDiagram

Правило	Множество имен	Множество имен правил
ElStep.In = ElBegin.Out, ElStep.Out = ElEnd.In		

После получения множества имен, необходимо его отфильтровать. Для этого используется целевое отфильтрованное множество, которое равняется разности Множества имен и Множества правил. С помощью данного фильтрования формируется квазитермальный алфавит блоков. Правилам присваиваются произвольные квазитерминальные символы (обычно a,b,c…), в процессе анализа.

Заключение

Метакомпиляция позволяет упростить задачу описания правил для RVTI-грамматики. Пользователь дается возможность описывать диаграмматический язык задав метаописания его графическим конструкциям, что ведет к более подробному описанию языка и обеспечивает полноту контроля топологии диаграмм.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417. Исследование выполнено при финансовой поддержке РФФИ и Правительства Ульяновской области в рамках научного проекта № 18-47-730032. Исследования поддержаны грантом Министерства образования и науки Российской Федерации, проект № 2.1615.2017/4.6.

Список литературы

1. Rekers J., Schurr A., Defining and visual parsing languages with layered graph grammars. Journal of Visual Languages and Computing, 8(1):27 –55, 1997.
2. GOLD Parsing System. Multi-Programming Language, Parser [Ýëåêòðîííûé ðåñóðñ]. – URL: http://goldparser. org/
3. Афанасьев А.Н., Гайнуллин Р.Ф. Метакомпилятор диаграмных языков // Автоматизация процессов управления. – С. 62-67.