

УДК 004, IS&IT'18

А.Н. Афанасьев¹, Н.Н. Войт², С.Ю. Кириллов³**МЕТОД СИНТЕЗА НА ОСНОВЕ ВРЕМЕННОЙ RV-ГРАММАТИКИ
ДЛЯ АВТОМАТНОГО МЕХАНИЗМА АНАЛИЗА И КОНТРОЛЯ
ДИАГРАММАТИЧЕСКИХ МОДЕЛЕЙ ДИНАМИЧЕСКИХ
РАСПРЕДЕЛЕННЫХ ПОТОКОВ РАБОТ**

Статья посвящена разработке метода метатрансляции для автоматного механизма анализа и контроля диаграмматических моделей динамических распределенных потоков работ, в основу которого заложена RVTI-грамматика. Предлагается система правил описания диаграмматических нотаций бизнес-процессов, на основе которой разрабатывается алгоритм синтеза автомата RVTI-грамматик, включая операции с внутренней памятью.

Потоки работ, диаграмматика, бизнес-процессы, метатрансляция, анализ визуальных языков.

A.N. Afanasyev, N.N. Voit, S.Yu. Kirillov

**SYNTHESIS METHOD BASED ON TEMPORAL RV-GRAMMAR FOR
AUTOMATIC MECHANISM OF ANALYSIS AND CONTROL OF
GRAMMATICAL MODELS OF DYNAMIC DISTRIBUTED WORK
FLOWS**

The article is devoted to the development of the method of getttranslatey for automatic mechanism analysis and control diagrammatically models of dynamic distributed workflows, which laid the basis for RVTI-grammar. A system of rules for describing business process diagram notations is proposed, on the basis of which an algorithm is developed for the synthesis of RVTI grammars, including operations with internal memory.

Workflows, diagrammatic, business process, meta-translation, the analysis of visual languages.

Введение

Диаграмматические модели, разработанные на визуальных языках (UML, IDEF, eEPC, BPMN, SharePoint, ER, DFD и др.) активно используются в практике проектирования сложных автоматизированных систем (ЦС), особенно на концептуальных этапах. При их построении есть веро-

¹ УлГТУ; г. Ульяновск, ул. Сев. Венец, 32; a.afanasev@ulstu.ru; 88422778846; д.т.н.; профессор.

² УлГТУ; г. Ульяновск, ул. Сев. Венец, 32; n.voit@ulstu.ru; 88422778846; к.т.н.; доцент.

³ УлГТУ; г. Ульяновск, ул. Сев. Венец, 32; kirillovsyu@gmail.com; 88422778846; начальник НИО ИДДО УлГТУ.

ятность совершения как синтаксических, так и семантических ошибок. Для их обнаружения существует множество методов и механизмов, одним из которых является семейство RV-грамматик [1-4]. Построение грамматики представляет собой прохождение нескольких этапов. Причем, для каждого диаграмматического языка строится своя грамматика. Это связано с различиями и особенностями самих языков. Эффективность данных грамматик была не раз доказана [5, 6]. Однако существует определенная сложность. Специалист, который хорошо разбирается в построении грамматики, может плохо знать особенности целевого визуального языка. Это наводит на то, что необходимы методы и инструменты, автоматизирующие синтез грамматики из определенного набора описаний нужного языка, который может составить знающий человек.

1. Алгоритм метатрансляции

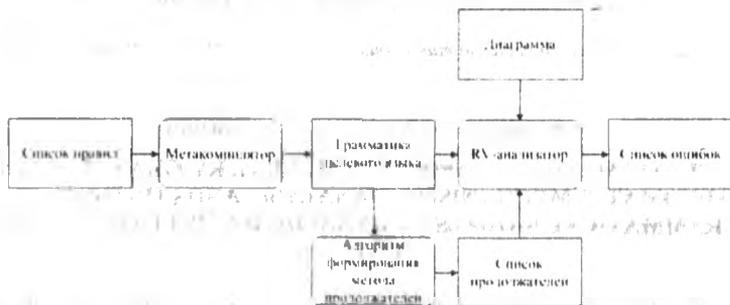


Рис. 1. Обобщенная схема синтеза и применения RV-анализатора

Процесс создания анализатора начинается с разработки правил графической грамматики. Метакомпилятор преобразует правила в таблицу грамматики целевого языка, понятную RV-анализатору. По данной таблице алгоритм формирования списка продолжателей строит множества пар продолжателей. По окончании этих этапов RV-анализатор готов к работе. На вход ему подаются анализируемая диаграмма, нужная грамматика и множество пар продолжателей для восстановления в случае нейтрализации ошибки. Результатом работы является список и местоположение ошибок, содержащихся в данной диаграмме.

2. Инструментальные средства метатрансляции

Для создания парсера или компилятора формального описания некоторого целевого языка используют программы класса метакомпиляторов. Подавляющее большинство современных программ данного класса – это

компиляторы парсеров. В качестве формального описания чаще всего используют запись языка в форме Бэкуса-Наура (БНФ) – формальная система описания синтаксиса, в которой одни синтаксические конструкции последовательно определяются через другие. Идеальный компилятор компиляторов по БНФ-описанию и множеству инструкций целевой архитектуры генерирует готовый к применению компилятор. На практике современные компиляторы компиляторов плохо интерпретируют семантику целевого языка, поэтому при проектировании компилятора компиляторов стоит ожидать только синтаксически верной интерпретации языка. YACC выбран в качестве примера, потому что является одним из универсальных инструментов и обладает полнотой свойств присущих инструментам его класса.

3. Разработка структуры метакомпилятора

Процесс метакомпиляции состоит из следующих этапов:

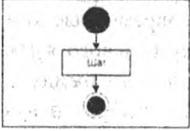
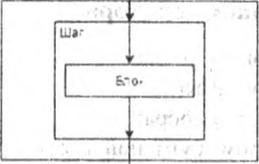
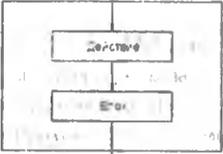
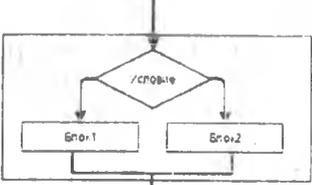
- 1) лексический разбор описания диаграммного языка;
- 2) синтаксический разбор и построение дерева разбора;
- 3) анализ дерева разбора и построение промежуточной структуры грамматики в терминах языка спецификаций;
- 4) трансляция – преобразование внутреннего представления в термины RV-грамматики;
- 5) оптимизация и минимизация;
- 6) сохранение таблиц RVTI-грамматики в формате XML.

В процессе анализа описания диаграммного языка метакомпилятор формирует набор паросочетаний между терминальными символами языка, строит таблицу переходов автомата RV-грамматики и генерирует набор операций с памятью для контроля соответствия вложенности нетерминальных символов и блоков с количеством входов или выходов больше одного.

«Отделение» описания RV-грамматики от ее анализатора позволяет использовать полученный XML-файл на разных платформах вне зависимости от того, с каким приложением интегрирован анализатор.

Описание спецификаций диаграммных языков состоит из правил, содержащих имя нетерминального символа, следующего после зарезервированного слова Rule, перечисления набора терминальных и нетерминальных символов, входящих в правило, а также описания отношений между терминальными и нетерминальными символами внутри правила. В табл. 1 приведены конструкции диаграмм активности и их метаописания.

Пример описаний для диаграммы активности UML

Графическое представление правила	Текстовое описание правила
	<p>Rule: IdDiagram Consist of: IdBegin ElBegin, IdSteps ElStep, IdEnd ElEnd Internal Relationships: ElStep.In = ElBegin.Out, ElStep.Out = ElEnd.In External Relationships: ;</p>
	<p>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;</p>
	<p>Rule: IdBlock Consist of: IdAction ElAction, IdBlock ElBlock Internal Relationships: ElBlock.In = ElAction.Out External Relationships: ElBlock.In = ElAction.In, IdBlock.Out = ElBlock.Out</p>
	<p>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</p>

	<p>Rule: IdBlock</p> <p>Consist of: IdBlock EIBlock, IdParal EIParal, IdJoinParal EIJJoinParal, IdCont EICont</p> <p>Internal Relationships: EIParal.Out1 = EIBlock.In, EIParal = EICont.In, EIJJoinParal = EICont.Out</p> <p>External Relationships: EIBlock.In = EIParal.In, EIBlock.Out = EIJJoinParal.Out</p>
	<p>Rule: IdCont</p> <p>Consist of: IdBlock EIBlock, IdParal EIParal, IdJoinParal EIJJoinParal, IdCont EICont</p> <p>Internal Relationships: EIParal.Out1 = EIBlock.In, EIParal = EICont.In, EIJJoinParal = EICont.Out</p> <p>External Relationships: EICont.In = EIParal, EICont.Out = IdJoinParal</p>

Синтез RVTI-грамматики производится в соответствии со следующими этапами:

- 1) построение квазитермального алфавита;
- 2) построение множества имен правил;
- 3) анализ количественных характеристик правил (выборка элементов, имеющих более одного входа или выхода);
- 4) построение операций с внутренней памятью.

На основе приведенных правил генерируются терминальные и квазитерминальные символы.

Квазитермальный алфавит строится на основе части правила, начинающегося с Consist of. Во время этого же прохода строится дополнительное множество имен правил.

После окончания первого прохода необходимо отфильтровать множество имен. Целевое отфильтрованное множество равно разности Множества имен и Множества правил. Таким образом формируется квазитермальный алфавит блоков.

Далее, перейдем к синтезу операций с внутренней памятью. У каждого правила «Блоки» есть вход и выход, следовательно, для контроля текста можно использовать стековый механизм. В начале обработки правила в стек заносится первый элемент. В конце обработки из стека извлекается этот элемент.

Заключение

Разработана система правил описания диаграмматических нотаций бизнес-процессов, на основе которой разработан алгоритм синтеза автомата RVTI-грамматик, включая операции с внутренней памятью. Дальней-

шие направления связаны с расширением возможностей семантического анализа диаграмматических моделей в плане согласования текстовой атрибутики диаграмм с проектной документацией.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 17-07-01417. Исследование выполнено при финансовой поддержке РФФИ и Ульяновской области в рамках научного проекта № 18-47-730032.

Библиографический список

1. **Афанасьев А.Н., Войт Н.Н.** Грамматико-алгебраический подход к анализу и синтезу диаграмматических моделей гибридных динамических потоков проектных работ // Информационно-измерительные и управляющие системы. – 2017. – № 12. – С. 69-78. – URL: <http://www.radiotec.ru/article/20138> (дата обращения: 25.05.2018)
2. **Афанасьев А.Н., Войт Н.Н., Уханова М.Е., ИONOва И.С., Епифанов В.В.** Анализ конструкторско-технологических потоков работ в условиях крупного радиотехнического предприятия // Радиотехника. – 2017. – № 6. – С. 49-58.
3. **Афанасьев А.Н., Шаров О.Г., Войт Н.Н.** Анализ и контроль диаграмматических моделей при проектировании сложных автоматизированных систем. – Ульяновск: УлГТУ, 2016. – 125 с.
4. **Afanasyev A.N., Voit N.N., Gainullin R.F.** Diagrammatic models processing in designing the complex automated systems //Application of Information and Communication Technologies (AICT), 2016 IEEE 10th International Conference on. – IEEE, 2016. – С. 1-4.
5. **Afanasyev A.N. et al.** Control of UML diagrams in designing automated systems software //Application of Information and Communication Technologies (AICT), 2015 9th International Conference on. – IEEE, 2015. – P. 285-288.
6. **Afanasyev A., Voit N., Gaynullin R.** The Analysis of Diagrammatic Models of Workflows in Design of the Complex Automated Systems //Proceedings of the First International Scientific Conference "Intelligent Information Technologies for Industry" (IIIT'16). – Springer International Publishing, 2016. – P. 227-236.