

ДЕДУКТИВНЫЙ ВЫВОД ОБЪЕКТНО-СОБЫТИЙНЫХ МОДЕЛЕЙ. ПРИМЕНЕНИЕ ПРИ РЕШЕНИИ ЗАДАЧ ДИНАМИКИ МНОГОФАЗНЫХ СРЕД

ПЕКУНОВ В.В., канд. техн. наук

Рассматривается проблема упрощения и частичной автоматизации разработки моделей для решения задач моделирования многофазных сред с применением системы порождения программ PGEN++. Предлагается использовать дедуктивный вывод фрагментов модели – элементов описания задачи и плана ее решения. Такой подход устраняет необходимость явного указания всех требуемых объектов и перечисления их комбинаций.

Ключевые слова: искусственный интеллект, целевые предикаты, схема дедуктивного вывода, автоматизация синтеза описаний задач моделирования.

DEDUCTIVE METHOD OF OBJECT-EVENT MODEL AND ITS APPLYING FOR PROBLEM SOLVING OF POLYPHASE MEDIUM DYNAMICS

V.V. PEKUNOV, Candidate of Engineering

The author considers the problem of simplification and partial automation of developing models to solve modulation problems of polyphase mediums with applying PGEN++ program system. The author suggests using the deductive inference of model fragments. They are problem describing elements and combination recitation.

Key words: artificial intelligence, object predicates, deductive inference diagram, automation of describing synthesis of modulation problems.

Качественное решение задачи автоматизации порождения программ моделирования многофазных сред требует построения подробной модели (описания) задачи, которая часто уже содержит базовый план решения в явной или неявной форме. Далее возможна либо прямая интерпретация модели (системы Флора/FloraWare), либо генерация решающей программы по структуре данной модели в соответствии со специализированной (системы HOPMA, SciNapse [1]) или одной из унифицированных схем: прямого преобразования (Draco, TAMPR [1], работы С.В. Востокина [2]); компилирующих шаблонов (MPS, системы класса Software Factories [1]); объектно-событийной интерпретации PGEN++ [3]. Построение модели задачи, содержащей план решения, – трудоемкая и нетривиальная проблема, для автоматизации решения которой часто используются технологии искусственного интеллекта, в частности, логический вывод (ПРИЗ [4], IPGS [5]) или последовательные символические преобразования.

Нами предлагается расширение гибкой схемы порождения программ на базе объектно-событийных моделей, реализованной в системе PGEN++, за счет частичного дедуктивного вывода плана решения задач. Пусть иерархия классов (понятий) предметной области включает порождающие и решающие (дедуктивные) классы. Задача описывается в виде графической схемы – совокупности взаимосвязанных объектов, являющихся экземплярами порождающих и/или решающих классов. Возможна даже исключительно формальная постановка задачи (в терминах «исходные данные – цель»), включающая лишь объекты решающих классов, которая в дальнейшем будет трансформироваться в развернутое описание задачи с фрагментами плана решения.

Структуру объектно-событийной модели, описывающей задачу, представим в виде динамического набора фактов, который взаимно однозначно отображается в XML-представление модели. Это позволяет применить технологии логического программирования как для анализа структуры задачи

(основанного на определении наличия или отсутствия соответствующих фактов), так и для ее трансформации (путем простой модификации базы фактов в результате логического вывода) в развернутую модель.

Пусть порождающие классы имеют генерирующие методы (на языке PHP), к каждому решающему классу относятся два предиката (на языке Prolog) – инициализирующий и решающий. Каждый предикат имеет входные (уже связанные) и выходные (свободные) переменные, состав которых определяется наборами входов и выходов, указанных в декларации соответствующего класса. Инициализирующие предикаты анализируют постановку задачи и подготавливают данные для решения, заполняя базу фактов, а решающие предикаты дедуцируют план решения и вводят его в описание задачи в виде соответствующей подсхемы, представляющей собой совокупность взаимосвязанных объектов. Интерпретация предикатов производится с помощью свободно распространяемого интерпретатора GNU Prolog.

Сначала задача рассматривается на уровне решения. Если модель задачи включает объекты решающих классов, к ней применяется процедура логического вывода, модифицирующая данную схему. Формируются два целевых предиката, требующих, соответственно, последовательного доказательства инициализирующих и решающих предикатов. Последовательность доказательства определяется структурой связей модели (по методике, эквивалентной сетевому графику работ). В результате доказательства целевых предикатов объекты порождающих классов генерируют подсхемы (фрагменты плана решения) и подставляют эти подсхемы вместо себя в модель. Если полученная модель вновь содержит объекты решающих классов, то вышеупомянутая процедура логического вывода повторяется.

К конечной расширенной модели (включающей план решения задачи), содержащей лишь объекты порождающих классов, применяется стандартная для системы PGEN++ [3] двухслойная интерпретация (на

событийном и объектно-сетевом уровнях), результатом которой является конечная программа, решающая задачу.

В настоящее время разработаны решающие классы для задач моделирования образования и распространения загрязнений, автоматизирующие синтез подсхем, вводящих в математическую модель блоки моделирования излучения. Это позволило существенно упростить описание задач моделирования в системе AirEcology-P, устранив необходимость явного указания всех требуемых объектов, их взаимосвязей и параметров, а также перечисления комбинаций объектов, описывающих те или иные особенности модели.

На рис. 1 показан пример фрагмента модели, содержащего объект порождающего класса «Излучение» и объекты решающих классов «Температура», «Капельная фаза» и «Химические реакции».

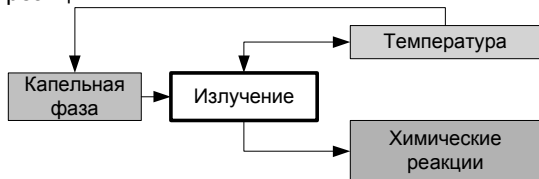


Рис. 1. Пример фрагмента исходной модели

В результате рассмотрения модели на уровне решения система определила, что в данном случае необходимо ввести (вместо исходного блока «Излучение») объекты, соответствующие трем диапазонам излучения, а также объекты, описывающие взаимодействие излучения с прочими компонентами модели (результат показан на рис. 2). Такой выбор был сделан на основании анализа исходной постановки задачи (рис. 1), разумеется, при иной структуре связей с блоком «Излучение» изменился бы и набор объектов, включаемых в модель.

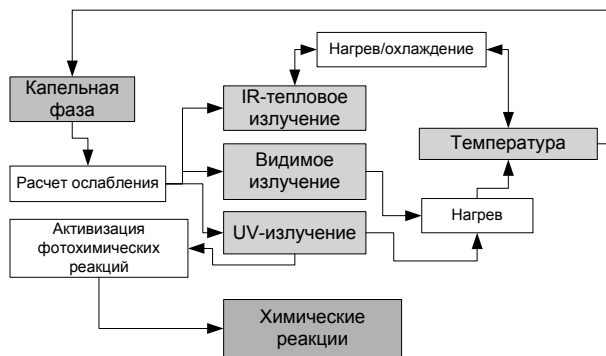


Рис. 2. Фрагмент исходной модели после применения процедуры логического вывода

Заключение

Предложенная схема дедуктивного вывода упрощает и частично автоматизирует разработку моделей, устраняя необходимость явного указания всех требуемых объектов и перечисления их комбинаций. В настоящее время такая схема используется для автоматизации синтеза описаний задач моделирования, учитывающих излучение и химическую кинетику в системе AirEcology-P.

Список литературы

1. Чарнецки К., Айзенкер У. Порождающее программирование: методы, инструменты, применение. – СПб.: Питер, 2005.
2. Востокин С.В. Технология визуального проектирования параллельных и распределенных приложений // Системы управления и информационные технологии. – 2006. – № 2(24). – С. 39–43.
3. Пекунов В.В. Автоматизация параллельного программирования при моделировании многофазных сред // Информационные технологии. – 2008. – № 5. – С. 37–42.
4. Кахро М.И., Калья А.П., Тыгу Э.Х. Инструментальная система программирования ЕС ЭВМ (ПРИЗ). – М.: Финансы и статистика, 1988.
5. Зубков В.П., Назаретский С.П. IPGS – интеллектуальная система автоматизированного программирования // Инф. среда вуза: сб. ст. – Иваново: ИГАСА, 2000. – С. 213–215.

Пекунов Владимир Викторович,
 ГОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,
 кандидат технических наук, доцент кафедры высокопроизводительных вычислительных систем,
 телефон (4932) 26-98-29.