

## Раздел VI. Вычислительные комплексы нового поколения и нейрокомпьютеры

УДК 004.4'4

Ю.В. Чернухин, В.Ф. Гузик, М.Ю. Поленов, А.А. Герасимов

### ПРИМЕНЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СРЕДСТВ ДЛЯ АВТОМАТИЗАЦИИ ПРОЦЕДУР МУЛЬТИЯЗЫКОВОЙ ТРАНСЛЯЦИИ С ЯЗЫКОВ МОДЕЛИРОВАНИЯ

*Рассматривается применение экспертной системы совместно с разработанной средой мультязыковой трансляции – Мультитранслятором для решения задачи автоматизации трансляции моделей, написанных на различных языках моделирования, в требуемый исследователю выходной формат представления на целевом языке. Необходимость применения дополнительных интеллектуальных средств вызвана определенными различиями в формальных схемах описания моделей, реализованных в современных языках и средах моделирования сложных систем. Использование данного подхода позволяет повысить эффективность процесса мультязыковой трансляции моделей при их импорте/экспорте для различных систем моделирования.*

*Многоязыковая трансляция; Мультитранслятор; экспертная система; языки моделирования.*

Yu.V. Chernukhin, V.Ph. Guzik, M.Yu. Polenov, A.A. Gerasimov

### APPLICATION OF THE INTELLECTUAL TOOLS FOR AUTOMATIZATION OF MULTILANGUAGE TRANSLATION FROM SIMULATION LANGUAGES

*Application of an expert system together with the developed multilanguage translation environment (software package Multitranslator) for automation of models' translation from various modeling languages into required to researcher output format on target language is considered. The necessity of additional intellectual tools usage is caused by defined differences in formal schemes of models' description implemented in modern languages and environments for complex system simulation. Usage of the given approach allows improving work of the multilanguage translation of models in their import/export for various simulation environments.*

*Multilanguage translation; Multitranslator; expert system; simulation languages.*

**Введение.** В процессе моделирования сложных мультидисциплинарных технических систем зачастую требуется привлечение специалистов, работающих в различных областях на разных языках и средах моделирования. Однако если для построения и исследования модели сложной системы необходима определенная среда и соответствующий ей язык моделирования, то возникает задача преобразования используемых моделей компонент к единому языку.

Для решения этой задачи целесообразно использовать инструментальные средства многоязыковой трансляции [1]. Одним из таких средств является разработанный программный комплекс «Мультитранслятор» (МТ) [1], позволяющий производить трансляцию моделей с различных языков моделирования, посредством описания правил преобразования в трансляционных модулях. Применение МТ для перевода моделей различных систем моделирования неоднократно показывало свою эффективность [2,3]. В то же время, при трансляции моделей иногда возникают проблемы, связанные с различием представления данных в языках описания транслируемых моделей.

Различают два основных метода в описании моделей: аналитическое и имитационное моделирование. Примером аналитического моделирования может служить непрерывное моделирование на языке ACSL (современное название acslx [4]). Однако, большинство языков моделирования (ЯМ) используют имитационный подход. К таким языкам можно отнести транзактно-ориентированные, например, SIMULA, GPSS, SIMAN, процессно-ориентированные – PEPA, событийно-ориентированные – SIMSCRIPT, интегрированные – Arena, Stratum, и др. [5].

При трансляции моделей, реализованных на языках, использующих различные методы их описания, недостаточно использования стандартных средств Мультитранслятора [1]. Разрешением данной проблемы является привлечение пользователя, принимающего необходимые решения на разных этапах трансляции при помощи Мастера трансляции, или применение некоторых интеллектуальных средств, которые могут заменить работу такого Мастера.

**Применение экспертных систем для разрешения проблем мультитрансляции языков моделирования.** Процесс мультитрансляции представляет собой генерацию для каждой конструкции или набора конструкций входного языка некоторой конструкции или набора конструкций выходного языка по правилам, описанным в трансляционном модуле (ТМ) Мультитранслятора. Генерация выходного кода производится в процессе построения синтаксического дерева входного языка. Если генерация выходного кода не может быть осуществлена последовательно по мере разбора синтаксических символов входного языка, то возможно применение, либо отложенного формирования генерируемого кода, либо вставка кода посредством текстовых меток [6].

В реализации Мастера трансляции использовалось разбиение данных исходного кода на отдельные массивы, для удобства дальнейшей обработки. Для их обработки Мультитранслятор позволяет использовать модули расширения и динамические подключаемые библиотеки с необходимыми функциями, реализация которых может быть осуществлена пользователем на различных языках программирования высокого уровня, что позволяет использовать для решения поставленной задачи большого количества сторонних программных средств.

В качестве примера возникающих при мультитрансляции проблем, рассмотрим трансляцию с языка моделирования (например, ACSL [4]) в выходную конструкцию (XML [7]). При переводе следует учитывать особенности исходного и выходного языков: различия в синтаксисе, представлении данных и конструкций, наличие специализированных переменных. Это связано с тем, что не все конструкции, поддерживаемые входным языком, могут быть переведены в соответствующие конструкции выходного языка.

Современный язык ACSL позволяет строить полную нелинейную поведенческую модель процесса или системы. При моделировании может использоваться большое число переменных блоков и функций, описывающих элементы системы [4]. При трансляции модели необходимо соблюдать логическую последовательность описания блоков, что не всегда однозначно возможно определить из-за различия в представлении данных. Помимо прочего, как видно из табл. 1, XML требует доопределения некоторых параметров, отсутствующих в описании ACSL.

Таблица 1

**Описание константы на ACSL и XML**

| ACSL             | XML  |
|------------------|--|
| CONSTANT a = 1.0 | <parameter default="1.0" name="a" nature="real" unit="" type="instance"><br><comment value="Constant" /> |

При разработке ТМ можно сохранить порядок выполнения действий, заменяя операторы и функции одного языка, на соответствующие операторы и функции другого языка. Недостаток такого подхода заключается в том, что разнятся не только операторы, но и способы описания процессов. Необходимо также учитывать, что у каждого ЯМ имеются специализированные переменные, которые должны игнорироваться для предотвращения некорректной трансляции. Именно по этой причине разумно использование Мастера, при помощи которого МТ запрашивает у пользователя в диалоговом режиме дополнительные параметры для перевода. На рис. 1 показан вид такого Мастера.

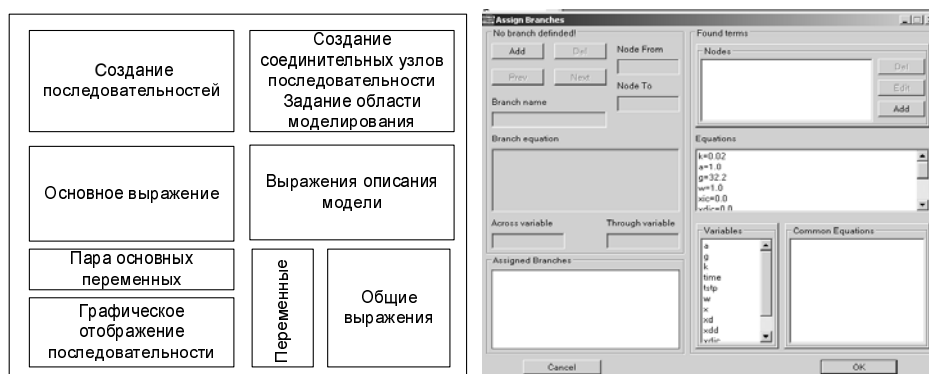


Рис. 1. Мастер трансляционного модуля ACSL-XML

Как видно из рисунка (см. рис. 1) пользователю приходится вручную определять последовательность действий в описании среды, области моделирования, размерности и типа элементов, распределения переменных и описаний, а также определения системных и общих параметров. Работать с таким Мастером может только пользователь-эксперт в области транслируемых языков, ведь неверное решение пользователя приведет к некорректному завершению трансляции.

В качестве альтернативного, более эффективного решения предлагается использование специализированной экспертной системы, выполняющей автоматически все функции пользователя-эксперта: хранение знаний в области моделирования и форматов моделей, принятие верных решений при возникновении неопределенностей, возможность, при необходимости, пополнения базы знаний [8].

Рассмотрим организацию взаимодействия работы Мультитранслятора с экспертной системой (рис. 2). После разбора входного кода, данные, поступавшие ранее Мастеру трансляции, передаются в рабочую память экспертной системы. Далее экспертная система (ЭС) начинает обработку поступивших данных. База знаний ЭС содержит правила обработки полученных данных.

Правила записываются на специальном языке символьного программирования и имеют вид: Если <Условие> тогда <Действия>. Правила могут иметь некую оценку, задаваемую при их описании или формируемую в процессе работы ЭС, что помогает при выборе варианта в ситуации неопределенности. При помощи механизма логического вывода правила поочередно применяются к полученным данным. Если условия применения правила удовлетворены, то происходит преобразование части данных в рабочей памяти, помечая их как обработанные.

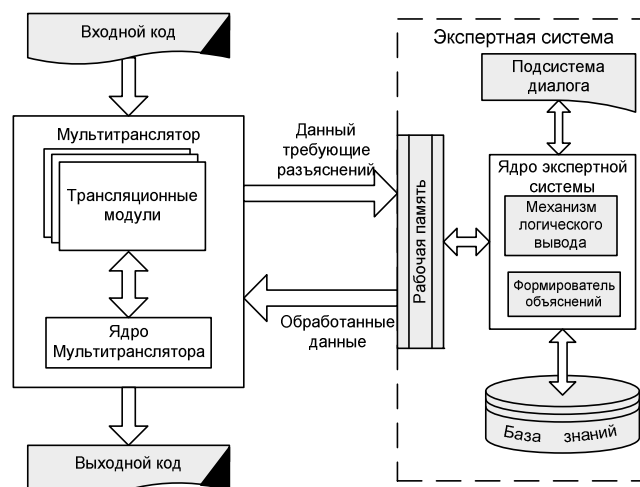


Рис. 2. Структура взаимодействия МТ и экспертной системы

Положительным результатом окончания работы ЭС является отсутствие не обработанных данных, и передача их трансляционному модулю Мультитранслятора для завершения формирования выходного кода. При ином результате производится дальнейший поиск необходимых правил. Ситуация при которой были применены все существующие правила, и остались непомеченные данные, говорит о том, что в базе знаний нет необходимых правил для корректного завершения работы ЭС, и требуется обучить систему введя новые правила для данной ситуации.

Процесс обучение сводится к заданию правил пользователем-экспертом. Большинство правил описано при создании ЭС, но для ее дообучения используется интеллектуальный редактор [8], являющийся частью подсистемы диалога. Недостающие правила могут относиться к разным этапам обработки данных, что позволяет оградить эксперта от тонкостей экспертной системы. Интеллектуальный редактор, зная какие данные не обработаны, предоставляет пользователю запрос на пополнение базы знаний, требующий знаний только в области моделирования.

Применяемые правила следуют в той или иной последовательности, благодаря чему формирователь объяснений может разъяснить пользователю почему были сделаны те или иные изменения. Данная функция не влияет на эффективность работы трансляции, но может помочь пользователю понять принцип работы ЭС.

**Интеллектуальная автоматизация трансляции моделей.** Работа экспертной системы должна замещать работу эксперта, следовательно, должна решать те же задачи и в той же последовательности.

Как видно из табл. 1 (см. табл. 1), ACSL не требует задавать константам ни единицы измерений, ни тип сигнала, требуемые в XML [7]. Применение экспертной системы позволяет определить «область моделирования» путем определения пары основных переменных through-across (см. рис. 2). Этот процесс проходит путем сравнения данных из поля с данными из ранее записанных в базу знаний экспертной системы (быстрый поиск). Если соответствий более одного, то вводится дополнительный параметр поиска – список наиболее часто используемых переменных в данной области моделирования (углубленный поиск). Разделение поиска на быстрый и углубленный позволяет ускорить работу ЭС. Благодаря такому поиску система может автоматически задать переменным единицы измерения и тип сигнала.

Блоки в ACSL-моделях оделены друг от друга операторными скобками [1], позволяющими определить количество блоков в системе и их границы. В данном примере приведен фрагмент описания модели пружины на языке ACSL, где операторные скобки: **DERIVATIVE** и **END**):

```
DERIVATIVE
time      = INTEG(1.0, 0.0)
xdd       = (w - k*xd - a*x) / (w/g)
xd        = INTEG(xdd, xdic)
x         = INTEG(xd , xic )
TERMT(t.GE.tstp, 'Time Limit')
END
```

После определения блоков модели экспертная система должна определить основное выражение блока – выражение, содержащее основные параметры. Для окончания трансляции необходимо исключить системные переменные, предварительно помещенные в базу знаний.

В данном случае задачей экспертной системы является интеллектуальный поиск необходимых конструкций с возможностью возврата для разбиения полученных данных на более сложные структуры.

Необходимую экспертную систему возможно создать при помощи специализированных средств, применяющихся для разработки ЭС. Примерами таких средств могут выступать: программная среда для разработки экспертных систем CLIPS, Prolog – язык логического программирования, РЕФАЛ – язык функционального программирования, LISP – язык обработки списков.

Среда CLIPS предлагает эвристические и процедурные подходы для представления знаний, допускает вызов внешних функций, написанных на других языках программирования; в свою очередь модули, написанные на CLIPS, могут быть вызваны программами, написанными на других языках [9], что позволяет ее совместное использование с МТ.

Средства CLIPS предоставляют возможность составлять определенную структуру правил по принятию решений, что позволяет в процессе разбора модели определять количество блоков, связь между блоками, и их внутренний состав. Так задача по разбору модели считается завершенной, если в рабочей памяти CLIPS не останется ни одного блока/параметра. После завершения разбора данные возвращаются в трансляционный модуль, который воспринимает их также как и данные при окончании работы мастера трансляции.

Таким образом ЭС, созданная на CLIPS, может использоваться в качестве интеллектуального средства автоматизации трансляции с языков моделирования совместно с средой Мультитранслятора.

**Заключение.** Представленный подход к автоматизации трансляции с языков моделирования также может быть применен для решения иных задач многоязыковой трансляции (например, для решения задач перевода моделей с универсальных языков программирования).

Необходимо заметить, что привлечение стороннего программного обеспечения является отнюдь не единственным способом построения ЭС. Существует возможность создания экспертной системы внутренними средствами Мультитранслятора, что должно повысить его функциональные возможности.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Чернухин Ю.В., Гузик В.Ф., Поленов М.Ю. Многоязыковая трансляция средств виртуального моделирования. – Ростов н/Д: Изд-во ЮНЦ РАН, 2009. – 368 с.

2. Чернухин Ю.В., Гузик В.Ф., Поленов М.Ю. Подход к формированию внешних библиотек сред виртуального моделирования на базе мультязыковой трансляции // Вестник компьютерных и информационных технологий. – 2008. – № 10. – С. 2-12.
3. Чернухин Ю.В., Поленов М.Ю., Булгаков Д.В. Использование мультязыковой трансляции при конверсии моделей, представленных на языках описания аппаратуры // Искусственный интеллект. – Донецк: Изд-во «Наука і освіта». 2009. – № 4. – С. 462-471.
4. *Моделирование с acslX*. Официальный сайт acslX. [Электронный ресурс]. Режим доступа: <http://www.acslx.com/products/modeling.shtml>
5. *Языки моделирования систем*. Claim – научно-образовательный кластер. [Электронный ресурс]. – Режим доступа: [http://www.philippovich.ru/Library/Books/ITS/wwwbook/ist5/glava\\_1/4\\_10.htm](http://www.philippovich.ru/Library/Books/ITS/wwwbook/ist5/glava_1/4_10.htm)
6. Чернухин Ю.В., Поленов М.Ю. Инструментальная подсистема многоязыковой трансляции виртуальных моделирующих систем // Известия ТРТУ. – 2004. – № 3 (38). – С. 115-120.
7. *Организация и функциональные возможности платформы XML*. Портал Российские электронные библиотеки [Электронный ресурс]. – Режим доступа: <http://www.elbib.ru/index.phtml?page=elbib/rus/methodology/xmlbase/analysis>.
8. *Коротко и понятно об экспертных системах*. Портал искусственного интеллекта, 2010. [Электронный ресурс]. – Режим доступа: <http://www.aiportal.ru/articles/expert-systems/short-about-expsys.html>.
9. Капустина Е.А. Применение средств языка CLIPS для организации и представления знаний в экспертных системах [Электронный ресурс]. – Режим доступа: <http://www.masters.donntu.edu.ua /2005/kita/kapustina/ind/clips.htm>.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

**Чернухин Юрий Викторович**

Технологический институт Федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге.

E-mail: [chernukhin@dce.tsure.ru](mailto:chernukhin@dce.tsure.ru).

347928, г. Таганрог, пер. Некрасовский, 44.

Тел.: 88634371656.

Кафедра вычислительной техники; профессор.

**Гузик Вячеслав Филиппович**

E-mail: [gvf@tsure.ru](mailto:gvf@tsure.ru).

Тел.: 88634371737.

Кафедра вычислительной техники; заведующий кафедрой.

**Поленов Максим Юрьевич**

E-mail: [polenov@dce.tsure.ru](mailto:polenov@dce.tsure.ru).

Тел.: 88634371656.

Кафедра вычислительной техники; доцент.

**Герасимов Александр Александрович**

E-mail: [sunsunger@gmail.com](mailto:sunsunger@gmail.com).

Тел.: 88634371656.

Кафедра вычислительной техники; аспирант.

**Chernukhin Yuri Victorovich**

Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”.

E-mail: [chernukhin@dce.tsure.ru](mailto:chernukhin@dce.tsure.ru).

44, Nekrasovskiy, Taganrog, 347928, Russia.

Phone: +78634371656.

The Department of Computer Engineering; Professor.

**Guzik Vyacheslav Philippovich**

E-mail: [gvf@tsure.ru](mailto:gvf@tsure.ru).

Phone: +78634371737.

The Department of Computer Engineering; Department Chair.

**Polenov Maxim Yuryevich**

E-mail: polenov@dce.tsure.ru.

Phone: +78634371656.

The Department of Computer Engineering; Associate Professor.

**Gerasimov Alexander Alexandrovich**

E-mail: sunsunger@gmail.com.

Phone: +78634371656.

The Department of Computer Engineering; Post-graduate Student.

УДК 004.27; 007.52

**А.Н. Берёза, М.В. Ляшов****ЭВОЛЮЦИОННЫЙ СИНТЕЗ КОНЕЧНЫХ АВТОМАТОВ\***

*Сложная цифровая система состоит из комбинационных и последовательностных логических схем, при этом поведение последовательностной схемы может быть описано конечным автоматом. Один из самых важных шагов в синтезе конечного автомата – это кодирование внутренних состояний. Задача кодирования внутренних состояний состоит в обнаружении лучшего назначения комбинаций триггеров к состояниям автомата. В современных системах автоматизированного проектирования для кодирования внутренних состояний конечного автомата используются алгоритмы NOVA, MUSTANG и их модификации. Однако сгенерированные коды состояний для конечного автомата являются далекими от оптимума. В данной статье рассмотрен разработанный эволюционный алгоритм кодирования внутренних состояний конечного автомата, показавший лучшие результаты по сравнению с традиционными методами.*

*Эволюционный синтез; NP-задача; конечный автомат; генетические алгоритмы.*

**A.N. Bereza, M.V. Lyashov****EVOLUTIONARY SYNTHESIS OF FINITE STATE MACHINES**

*A complex digital system consists of combinational and sequential logic circuits, and the behavior of sequential circuits can be described by a finite state machines. One of the most important steps in the synthesis of finite state machines is a state assignment. The state assignment task is searching the combinations triggers to the finite state machines. In modern systems, synthesis of coding the internal states of a finite automaton algorithms are used NOVA, MUSTANG and their modifications. However, state assignment of finite state machines is far from optimum. This article describes an evolutionary algorithm developed by coding the internal states of a finite state machines, which showed better results compared to traditional methods.*

*Evolutionary synthesis; NP task; finite state machines; genetic algorithms.*

**Введение.** Сложная цифровая система состоит из комбинационных и последовательностных логических схем, при этом поведение последовательностной схемы может быть описано конечным автоматом (КА). Методы синтеза КА непрерывно развиваются [1–5]. Один из самых важных шагов в синтезе КА – это кодирование внутренних состояний. Задача кодирования внутренних состояний состоит в обнаружении лучшего назначения комбинаций триггеров к состояниям

---

\* Работа выполнена при поддержке: АВЦП (гранты № 2.1.2/9625, № 2.1.2/9981), г/б № Г-11.11.МР (0120.1155814).