

ВЫБОР ФУНКЦИИ ПРИСПОСОБЛЕННОСТИ ДЛЯ АВТОМАТИЧЕСКОЙ ГЕНЕРАЦИИ СВЯЗЕЙ ДАННЫХ В ПРОГРАММАХ ИЗ ФУНКЦИОНАЛЬНЫХ БЛОКОВ

Миронович В. А., аспирант кафедры компьютерных технологий
Университета ИТМО, mironovich.vladimir@gmail.com

Буздалов М. В., к. т. н., доцент кафедры компьютерных технологий
Университета ИТМО, mbuzdalov@gmail.com

Аннотация

Согласно некоторым исследованиям – более 50% затрат в промышленных системах управления уходит на разработку и поддержку программного обеспечения (ПО). Автоматическая разработка программного обеспечения может позволить уменьшить затраты, упростить процесс разработки и увеличить качество ПО. Одним из подходов к автоматической генерации ПО является генетическое программирование. В данной работе предложен эволюционный подход к автоматической генерации программ из функциональных блоков стандарта IEC 61499 для промышленных систем управления. В работе рассмотрена задача генерации связей данных в приложении из функциональных блоков. На данной задаче произведена оценка применимости функции приспособленности, использующей число выполненных формул логики деревьев вычислений, а также произведен простой экспериментальный анализ производительности простого эволюционного алгоритма – (1+1).

Введение

Международный стандарт IEC 61499 [2] определяет архитектуру распределенных систем управления и автоматизации. Стандарт рассчитан на модульную разработку с повторным использованием компонент – функциональных блоков. Система управления определяется как сеть нескольких функциональных блоков. Блок определяет поведение и состояние некоторого микроконтроллера в системе. Он взаимодействует с другими блоками в системе через определенный интерфейс,

определяющий входные и выходные переменные и события. Реализация функциональных блоков основана на диаграммах управления выполнением (ЕСС), также называемых автоматами Мура. Для каждого состояния определено действие – алгоритм на языке Structured Text, представленном в стандарте [1]. Алгоритмы определяют выходные переменные и отправляют события, которые в свою очередь вызывают переходы в других блоках приложения.

Эволюционные вычисления [3, 6] – это общее название методов оптимизации вдохновленных идеями эволюции из биологии. Эволюционные алгоритмы используют различные методы, такие как мутация, скрещивание и селекция, для поиска оптимального решения, определяемого функцией приспособленности особи. Данные методы применимы к множеству задач оптимизации, и часто используются для решения таких задач, для которых не существует точных алгоритмов работающих на полиномиальное время.

В данной работе мы предлагаем метод автоматической генерации программ стандарта IEC 61499, и оцениваем эффективность функции приспособленности особи, определяемой числом выполненных формул в верификаторе UPPAAL [4]. В UPPAAL система моделируется с помощью временных автоматов [5]. Для верификации системы используется специальный язык запросов, являющийся подмножеством логики деревьев вычислений. Запрос состоит из формул состояния и формулы путей. Формулы состояния описывают требования к значениям переменных в состояниях, в то время как формулы пути описывают как формула состояния соотносится с поведением модели во времени.

Предложенный метод

Мы предлагаем и производим оценку применимости метода проверки моделей и эволюционных алгоритмов для автоматической генерации приложений из функциональных блоков стандарта IEC 61499. В качестве исходной задачи мы рассматриваем генерации связей данных в приложении, в случае когда используемые функциональные блоки и связи событий уже установлены.

Интерфейс функционального блока определяет входы и выходы для переменных и данных. У блока, представленного на Рис 1, для событий определен один вход REQ и два выхода INITTO, IND. Для данных определено три входа PV1, PV2 и LABEL, а также один выход POSITION.

Предложенный метод автоматически генерирует связи данных между функциональными блоками используя эволюционные алгоритмы

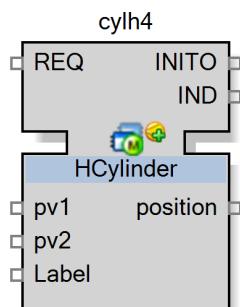


Рис. 1: Пример функционального блока

и функцию приспособленности на основе запросов в верификаторе UPPAAL. Полученное в результате приложение должно выполнять требования спецификации и реализовывать необходимый функционал.

Мы производим оценку метода на простой системе, представленной на Рис. 2 (слева). Данная система является частью автоматизированного манипулятора, с системой контроля построенной согласно архитектуре Интернет Вещей: каждый клапан и сенсор имеет собственный микроконтроллер. Система состоит из одного цилиндра, двух клапанов и датчика положения. Требуемая позиция поступает с координаты X джойстика. Приложение из функциональных блоков, представленное на Рис. 2 (справа) представляет систему замкнутого цикла, в которой функциональные блоки PopH1 и PushH1 управляют клапанами, а joystick4 является интерфейсом джойстика, задающим координату X. Функциональный блок cylh4 является интерфейсом модели поведения цилиндра. В этой системе 7 выходов данных и 8 входов, каждый выход может быть соединен с неограниченным числом входов. Цель автоматической генерации связей данных – определить, к каким выходам подключается каждый из блоков.

Для заданной системы была разработана модель в верификаторе UPPAAL приведенная на Рис. 3. Для каждого из вышеописанных блоков, кроме модели джойстика, был разработан временной автомат, сеть которых моделирует поведение полной системы. Поведение джойстика заменено константными значениями X. Все выходные переменные исходной модели представлены в виде глобальных переменных, передаваемых в каждый из автоматов через указатель. Это позволяет смоделировать связь данных в исходной программе, и использовать метод проверки моделей для оценки корректности генерируемых связей.

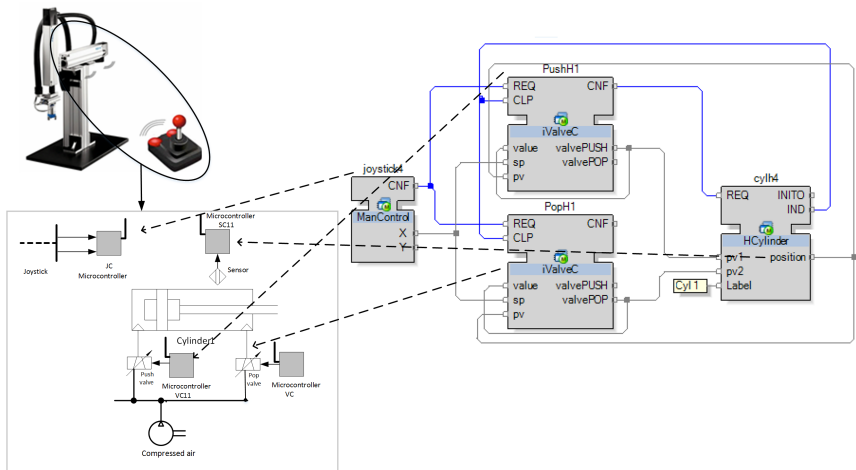


Рис. 2: Распределенная система управления цилиндром

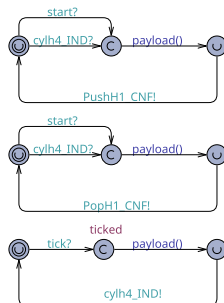


Рис. 3: Модель приложения в UPPAAL

Итоговая задача оптимизации определена следующим образом: в заданной модели UPPAAL необходимо для каждого входа данных определить к какому выходу он должен быть подключен для обеспечения полного соответствия модели заданной спецификации. Функция приспособленности – число выполненных запросов CTL.

Построенная в UPPAAL модель реализовывала поведение системы “передвинуть цилиндр из исходного положения $x = 0$ в положение заданное джойстиком X ”. Для данного поведения было написано шесть формул, описывающих как требования к состоянию системы в целом,

так и конкретные требования к поведению. Функция приспособленности особи может принимать значения от 0 до 6.

Особью алгоритма оптимизации является модель с переменными соединениями данных. Каждая особь задается восемью переменными – входами, принимающими значения от 0 до 6. Первые два значения определяют входы блока `culh4`, следующие три – `PushH1` и последние три – `PopH1`. Исходное приложение может быть задано вектором [2, 5, 2, 0, 6, 5, 0, 6].

Результаты

Для оценки функции приспособленности мы сгенерировали и про- верифицировали 5 764 801 возможных моделей. Распределение значений функции приспособленности представлено в Таблице 1. Из 5 764 801 особей только 392 удовлетворили все 6 рассмотренных запросов. Все оптимальные решения соответствуют следующему шаблону: $\{\{0, 2, 4\}; \{1, 3, 5\}; *, 0; 6; *, 0; 6\}$. Функция приспособленности правильно определяет соединения для моделей клапанов – $[*, 0; 6]$. Первая переменная отвечает за визуализацию, и не влияет на работу модели. Оставшиеся переменные принимают на вход положение джойстика (0) и позицию цилиндра 6. Это критично для поведения модели и одинаково для всех оптимальных моделей. Первые две переменные, отвечающие за вход цилиндра, принимают на вход либо положение джойстика, либо необходимые значения с клапанов. Так как клапаны имеют одинаковую логику – их выводы одинаковы, поэтому значения 2, 4 и 3, 5 взаимозаменяемы. Доработка модели позволила избавиться от использования положения джойстика в этих переменных. Итоговый шаблон оптимума – $\{\{2, 4\}; \{3, 5\}; *, 0; 6; *, 0; 6\}$, что соответствует правильным значениям необходимым для работы модели и приложения.

Мы также провели оценку работоспособности простого эволюционного алгоритма – (1+1) на данной задаче. Мы произвели 300 запусков алгоритма используя рассматриваемую функцию приспособленности. На Рис. 4 представлены график числа вычислений функции приспособленности необходимых для достижения оптимум. Медиана числа итераций – 10 796, однако их значения разнятся от 5 до более 80 000. Следует отметить, что хотя верификация правильной модели почти не занимает времени, для некорректных связей верификатор может затратить много времени на верификацию, или не найти его в принципе.

Значение	Число моделей	Доля
0	3818	0.066%
1	159341	2.764%
2	1792404	31.092%
3	3477201	60.318%
4	286226	4.965%
5	45419	0.788%
6	392	0.007%
Всего	5764801	100%

Таблица 1: Распределение значений функции приспособленности

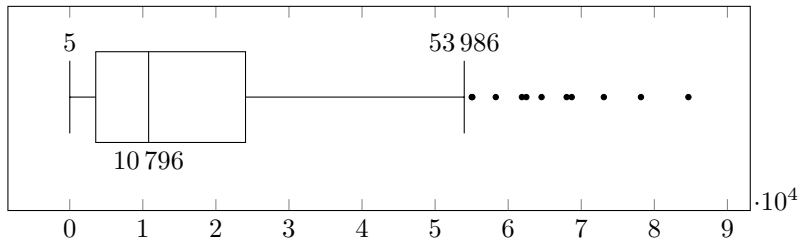


Рис. 4: Число вычислений функции приспособленности необходимое для достижения оптимума

Заключение

Мы представили метод автоматической генерации программ из функциональных блоков использующий эволюционные алгоритмы и проверку моделей. В данной работе мы рассмотрели оценили возможность использовать функцию приспособленности основанную на числе выполненных запросов верификатора UPPAAL.

Используемая функция приспособленности верно определяет необходимые связи данных в приложении, и позволяет использовать эволюционные алгоритмы (на примере алгоритма (1+1)). В дальнейшем, данный подход можно расширить на генерацию связей событий, и приложений в целом, а также использовать другие средства верификации и автоматические переводчики приложений IEC 61499 в модели для верификаторов.

Литература

- [1] International Standard IEC 61131-3: Programmable controllers – Part 3: Programming languages, 2nd ed. // International Electrotechnical Commission, Geneva, 2003.
- [2] International Standard IEC 61499-1: Function Blocks – Part 1: Architecture, 2nd ed. // International Electrotechnical Commission, Geneva, 2012.
- [3] T. Bäck. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. // Oxford University Press, Oxford, UK, 1996.
- [4] G. Behrmann, A. David, K. G. Larsen, J. Hakansson, P. Petterson, W. Yi, and M. Hendriks. Uppaal 4.0. // Third International Conference on the Quantitative Evaluation of Systems - (QEST'06), pages 125–126, Sept 2006.
- [5] J. E. Hopcroft, R. Motwani, and J. D. Ullman. Introduction to Automata Theory, Languages, and Computation (3rd Edition). // Addison Wesley, 3 edition, July 2006.
- [6] Ю.А. Скобцов. Основы эволюционных вычислений. // ДонНТУ, Донецк, 2008.