ИССЛЕДОВАНИЕ УСЛОВИЙ ПРИМЕНИМОСТИ ЯЗЫКА ПАРАЛЛЕЛЬНОГО ПРОГРАММИРОВАНИЯ СПАРМ ДЛЯ ЗАДАЧ ПОСТРОЕНИЯ НАДЕЖНЫХ УПРАВЛЯЮЩИХ ПРОГРАММ [1]

В.Е. Зюбин
Инженерный Центр
Института Автоматики и Электрометрии
Сибирского отделения РАН
630090 Новосибирск, просп. им. ак. Коптюга, 1
Россия
E-mail: zyubin@iae.nsk.su
Тел. (3832)399421

Ключевые слова: SPARM language, safety critical control systems, safety criteria, complex algorithms, human factor, reliability, robustness, traceability, maintainability

Abstract

This paper presents research results on use of the SPARM language for operating algorithms of safety-critical control systems. Safety-related criteria and safety-related features of the language are discussed.

1. Введение

Возрастающая сложность технических систем и проблема их автоматизации диктуют необходимость создания принципиально новых методик проектирования. Ключевым требованием, которое предъявляется к таким методикам, является требование надежности создаваемого программного обеспечения(ПО), что особенно актуально для класса задач автоматизации "критичных" объектов на предприятиях металлургической и химической промышленности, атомных электростанциях, предприятиях космической индустрии, а также класса задач автоматизации сложных научно-технических экспериментов, связанных с такими производствами. Для перечисленных случаев возникновение ошибки не только влечет неприемлемые материальные и финансовые потери, но зачастую имеет катастрофические последствия - вплоть до человеческих жертв.

В большинстве случаев надежность языкового средства трактуется как мера степени обнаружения ошибок на этапе трансляции и исполнения программы. Однако, для любой языковой системы имеется предел обнаруживаемых ею ошибок. Например, если распознаваемость семантически ошибочных языковых конструкций может быть повышена за счет типизации переменных, то ошибки в логическом построении программы не поддаются автоматическому обнаружению [1]. При этом, чем более сложным является объект управления, тем более вероятно возникновение логических ошибок. Обычно в качестве основных факторов, повышающих надежность создаваемой программы, называются такие характеристики как удобочитаемость и простота. А в качестве средств их достижения - степень типизации переменных, вид языковых конструкций и способ формировании листинга программы [1, 2, 3]. Перечисленные факторы могут быть учтены на этапе определения синтаксиса языка, однако проблема формирования логической

---

[1] Ссылка на статью:
Зюбин В.Е. Исследование условий применимости языка параллельного программирования СПАРМ для задач построения надежных управляющих программ // Распределенная обработка информации. DDP'98: Тр./Шестой международный семинар. – 23-25 июня, 1998, Академгородок, Новосибирск. С.122-126.

структуры алгоритма требует отдельного рассмотрения. Этот неразрешимый для языков общего назначения вопрос в специализированных языках может быть частично снят за счет ориентации средства программирования на конкретную методологию.

В настоящей работе рассматриваются характеристики созданного ранее языка параллельного программирования СПАРМ (Средство Программирования Алгоритмов Работы Микроконтроллеров) на соответствие надежностным критериям, предъявляемым к языковым средствам.

2. Критерии надежности и методология

Одной из последних работ, посвященных выработке критериев надежности (safety) ПО и систем его разработки, является [4], финансированная Nuclear Regulatory Commission. В этой работе критерии, связанные с надежностью ПО, разделены на четыре категории: а) **Reliability (**вопросы устойчивого функционирования ПО в "нормальных" (определенных на этапе создания технического задания) условиях); б) **Robustness** (вопросы устойчивого функционирования ПО при исключительных ситуациях и аварийных событиях); в) **Traceability** (вопросы, касающиеся организации исходного текста и библиотек, их качества и возможности сверки на соответствие техническому заданию); г) **Maintainability** (вопросы модифицируемости исходного текста в процессе эксплуатации).

Эта работа была принята в качестве основы для проводимого исследования. При этом критерии каждой из категорий были дополнительно структурированы по признаку связанности с человеческим фактором - фактором, очевидно присутствующим при создании программного обеспечения, но зачастую либо не выделяемого из системы "программист - компьютер", либо не рассмаривемого вовсе. Это достаточно принципиальное с точки зрения автора дополнение позволило упростить рассмотрение проблемы за счет разделения круга вопросов на две группы:

а) группу факторов, связанных с психологическими аспектами процесса проектирования (минимизация смешано-языкового программирования, четкая структуризация, простота, информационная наполненность исходного текста, и т.п.);

б) группу факторов, связанных с аппаратно-программной реализацией алгоритма (минимизация прерываний, недопустимость рекурсивных вызовов функций и динамических операций с ОЗУ, нежелательность использования многозадачной модели параллелизма и т.п.).

Очевидно, что обеспечение большинства требований первой группы не может быть гарантировано формальными языковыми средствами, например, ввиду ограничений, рассмотренных в работах Геделя, Тарского, Черча [5]; языковое средство в состоянии лишь способствовать и предоставлять принципиальную возможность выполнения этих требований за счет своей методологической ориентации. Вторая же группа вопросов может быть строго удовлетворена на этапе разработки и реализации языкового средства. Каждая из этих групп допускает рассмотрение едиными блоками, что позволяет в сжатом виде осветить ключевые характеристики языкового средства.

3. Базовые характеристики языка СПАРМ

СПАРМ является специализированным языком параллельного программирования, ориентированным на описание алгоритмов функционирования сложных автоматизированных или полностью автоматических систем. В качестве математической модели дискретного устройства в СПАРМ используется H-автомат (гипер-автомат), т.е. совокупность параллельно-исполняемых слабосвязанных Z-автоматов(или процессов), которые в свою очередь являются расширением классического конечного автомата. Процессы исполняются параллельно. Процессы можно останавливать, приостанавливать на заданное время, запускать с начала, запускать с состояния, в котором он был остановлен, запускать с произвольного состояния. Из каждого процесса можно получить информацию о состоянии любого другого процесса и изменить это состояние.

Процессы равноправны. Для организации параллельного исполнения выбрана мультипоточная модель параллелизма. СПАРМ имеет текстовый русскоязычный синтаксис и ориентирован на язык Си в части формы описания выражений и встроенного интерфейса с этим языком. Эквивалентное отображение описания алгоритма в машинный код также базируется на языке Си. В силу специфических особенностей СПАРМ (параллелизм, модифицированная типизация переменных и необходимость контроля этой типизации) эквивалентное преобразование осуществляется трансляционной моделью. Переносимость обеспечивается выделением аппаратно-зависимых процедур ввода-вывода [6].

4. Неформализуемые (человеко-зависимые) свойства языка

Человеко-зависимые аспекты программирования концентрируются вокруг неформализуемых вопросов структуризации ПО, его оптимизации и простоты. Степень соответствия СПАРМ этим требованиям определяется базовыми концепциями СПАРМ, заложеными на этапе разработки языка. В частности, это:

а) простота и общность правил написания языковых конструкций, базирующихся на Си;

б) ориентация на русскоязычный синтаксис и предоставление возможности создания полноценных идентификаторов;

в) возможность идентификационной развязки информационной наполненности алгоритма на уровнях:

- аппаратуры (физических портов ввода/вывода);
- временного течения алгоритма (памяти алгоритма);
- элементарных исполнительных устройств (клапаны, агрегаты, датчики и т.п.);
- организации произвольных, в том числе и замкнутых, иерархических структур, не ограничивающих количество уровней структуризации алгоритма и обеспечивающих отражение функциональной наполненности алгоритма по процессуальному признаку.

Последнее из перечисленного позволяет рассматривать создание ПО системы управления в языковой среде СПАРМ как отражение процесса разработки автоматизируемого объекта (как такового), т.е. ставит психологические предпосылки разработки надежного ПО в зависимость от методологии, использованной при конструировании самого устройства. Остальные свойства СПАРМ предоставляют средства для наиболее эффективного достижения этой цели. В совокупности с базовыми характеристиками языка этим достигается удовлетворение следующих требований по надежности ПО рассматриваемого класса задач:

а) максимальная степень структуризации алгоритма при минимизации перекрестных логических связей (обеспечивается ориентацией СПАРМ на структуризацию алгоритма по процедуральному признаку, соответствующего природе задач управления. Как следствие, предоставление возможности разбиения сложных алгоритмов управления на модули, удовлетворяющие требованию самодостаточности (selfishness, часто обозначаемой терминами "скрытие информации" или "инкапсуляция"));

б) отсутствие множественности в точках входа и выхода Z-автоматов (обусловливается свойствами процессов);

в) локализацию обработки исключительных ситуаций непосредственно в точках их обнаружения (равноправность процессов и отсутствие жестких родственных отношений позволяет изменять нормальное течение алгоритма процессом, расположенным на произвольном иерархическом уровне);

г) унифицированную обработку исключительных ситуаций на уровне процессов (определяется унифицированной природой Z-автомата);

д) удовлетворительное отображение алгоритма, текстурально заданного документацией (текстовая форма описания, русскоязычный синтаксис);

е) минимизацию смешанно-языкового программирования (использование встроенного интерфейса с языком Си в экстраординарных случаях его применения, с одной стороны, предполагает сохранение вида описания выражений, а, с другой стороны,

обеспечивает либо концентрацию этого инородного кода в отдельных модулях, либо его маркирование в тексте программы);

ж) концентрацию связанного кода и данных внутри процесса;

з) высокую модифицируемость программ.

Особый интерес представляет обсуждение текстового вида описания программ, принятого в СПАРМ. Текстовый вид описания на первый взгляд проигрывает графической форме, которая становится все более популярной в связи с широким распространением графических интерфейсов операционных систем семейства Windows, и часто воспринимается как некоторый недостаток. Однако, анализ психологических аспектов проблемы [7] показывает, что в первую очередь ответ на вопрос должен быть связан с ассоциативными характеристиками языковых средств. В этом случае, констатируя невозможность создания четких правил конструирования новых символов иероглифическими средствами, можно сделать однозначный вывод о непригодности их использования для организации новых уровней абстракции, неизбежно возникающих при описании работы сложных объектов автоматизации; и, как следствие, - вывод о правомерности предлагаемого подхода.

При этом надо отметить, что перечисленная часть свойств языка не снимает требований к квалификации программиста и дисциплинарной организации процесса разработки ПО в части дополнительных комментариев, выработке системы построения идентификаторов, глубины разбиения алгоритма, использования литеральных констант, качества программной документации и т.п., которые находятся в экспоненциальной зависимости от сложности проекта.

К недостаткам текущей версии языка необходимо отнести отсутствие механизма макроописаний и конструкций многовариантного переключения.

5. Аппаратно-зависимые свойства языка

Вид реализации СПАРМ ориентирует использование языка для встроенных системно-независимых приложений, а мультипоточная организация параллелизма позволяет исключить из рассмотрения весомую часть вопросов, касающихся аппаратно-зависимых критериев надежности: а) вопросы динамического использования ОЗУ и своппинга; б) вопрос динамических межмодульных связей; и в) вопрос минимизации многозадачности.

Язык Си, на который ориентирован СПАРМ, несмотря на ряд несомненных достоинств (повышенная переносимость, широкое распространение, гибкость и т.д.), общепризнанно является языком мало подходящим для построения надежного ПО. Это справедливое отношение к Си обусловлено недостаточной степенью типизации переменных, возможностью неконтролируемой передачи управления и использования средств динамической работы с памятью. Для устранения этих опасных с точки зрения надежностных характеристик ПО факторов транслятор СПАРМ несет дополнительную функциональную нагрузку: контроль жесткой типизации переменных, обусловленной синтаксисом СПАРМ; контроль корректности управления процессами; и выявление "мертвых" (не имеющих точек выхода) состояний. Отсутствие операций динамической работы с памятью и операций передачи управления, отличных от контролируемых, обеспечено конструктивно - организацией Z-автомата

Временная предсказуемость (существование заведомо конечного времени реакции системы управления на внешнее событие) обусловлена реализацией параллелизма в рамках мультипоточной модели. Это позволяет еще на этапе проектирования алгоритма определить максимальное время реакции. Максимальное гарантированное время реакции системы на внешнее событие ($T_{макс}$) определяется временем $T_{накл}$, затраченным на накладные расходы по организации функционирования алгоритма (связь с физическими устройствами ввода/вывода, организация параллельного исполнения), и временем, затраченным собственно на выполнение алгоритма ($T_{алг}$). При этом формула для $T_{накл}$ имеет достаточно простой вид и не вызывает проблем при анализе.

$T_{накл} = N_{пр}*T_{пр} + \Sigma(N_{вхфm}*T_{вхфm}) + \Sigma(N_{вхj}*T_{вхj}) + \Sigma(N_{выхk}*T_{выхk}) + \Sigma(N_{выхфl}*T_{выхфl})$, где

$N_{пр}$ - число процессов;

$T_{пр}$ - время, затраченное на обеспечение параллелизма исполнения процесса;

$N_{вхфm}$ - число входных физических портов разрядности m;

$T_{вхфm}$ - время, затраченное на считывание одного входного физического порта разрядности m;

$N_{вхj}$ - число входных программных переменных типа j;

$T_{вхj}$ - время, затраченное на преобразование от физического представления входной переменной типа j к программному;

$N_{выхk}$ - число выходных программных переменных типа k;

$T_{выхk}$ - время, затраченное на преобразование выходной переменной типа k от программному представления к физическому;

$N_{выхфl}$ - число выходных физических портов разрядности l;

$T_{выхфl}$ - время, затраченное на запись одного выходного физического порта разрядности l.

Вычисление $T_{алг}$, если ввести предположение об одновременном исполнении всех процессов, также не имеет принципиальных препятствий для вычисления.

$T_{алг} = \Sigma\ Max(P_n)$, где

$Max(P_n)$ - время исполнения состояния процесса n, содержащего наиболее ресурсопотребляющий по времени набор инструкций.

Таким образом, искомое максимальное гарантированное время реакции равно

$T_{макс} = 2*(T_{накл} + T_{алг})/(1 - K)$, где

$K \in [0, 1]$ - доля вычислительной мощности целевой системы, затраченная на обработку высокочастотных сигналов, обрабатываемых по прерываниям, при их максимально возможной частоте поступления (К = 1 означает что выбранная целевая система будет заниматься только обработкой таких сигналов).

Ввиду того, что $T_{алг}$ вычисляется для случая выполнения всеми процессами своих наиболее ресурсопотребляющих состояний, реально наблюдаемое время реакции системы будет

$T_{реал} \in (T_{макс}, T_{накл})$.

СПАРМ ориентирован на встроенные приложения, что позволяет создавать управляющие программы без дополнительного привлечения средств операционных систем, что снимает проблему использования "инородного" (неконтролируемого разработчиком ПО) кода.

Необходимость жесткой привязки течения алгоритма ко времени предусматривает использование единственного прерывания - прерывания от таймера.

Переносимость программ обеспечена свойствами языка Си и сводится к коррекции десятка выделенных аппаратно-зависимых процедур и физической адресации портов. Описание функциональной наполненности алгоритма, выполненное штатными средствами СПАРМ, при этом не изменяется.

6. Заключение

Анализ базовых характеристик языка СПАРМ на соответствие требованиям надежности позволяет сделать заключение о допустимости его применения в задачах автоматизации сложных объектов критичных производств.

Список литературы

[1] Янг С. Алгоритмические языки реального времени: конструирование и разработка. М.: Мир, 1985.

[2] Кнут Д. Искусство программирования для ЭВМ. М.: Мир, 1976.

[3] Хоор К. О структурной организации данных. В кн.: Дал У., Дейкстра Э., Хоор К. Структурное программирование. - М.: Мир, 1978.

[4] Guidlines on Software Languages for use in Nuclear Power Plant Safety Systems/ Decker D., Dinsmore G., Graff S., Green W., Hecht H., Justice M., Koch S., Lin D., Ossia K., Pollard J., Shokri E., Sorkin A., Tai A., Tso K.S., Wendelboe D. - NRC, 1997. http://www.nrc.gov/NRC/NURREGS/SR6463/index.htm

[5] Одинцов Б.Е., Дик В.В. Синтаксичность моделей баз знаний интеллектуальных систем // Приборы и системы управления. 1998, N1. С.15-17

[6] Зюбин В.Е. Язык СПАРМ - средство программирования микроконтроллеров// Автометрия, 1996, N2. С. 40-50

[7] Ericsson A., Kintsch W. "Long-Term Working Memory", 1996. http://cogsci.soton.ac.uk/~harnad/Papers/Py104/ericsson.long.html