УДК 004.415

В.С. ХАРЧЕНКО[1], В.В. СКЛЯР[1], А.А. ГОРДЕЕВ[1],
В.И. ТОКАРЕВ[2], А.Д. ГЕРАСИМЕНКО[2], Ю.А. БЕЛЫЙ[2]

[1] Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина
[2] ЗАО «Радий» (г. Кировоград), Украина

ИСПОЛЬЗОВАНИЕ МЕТРИК ХОЛСТЕДА ПРИ ОЦЕНКЕ БЕЗОПАСНОСТИ КРИТИЧЕСКОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Предложена методика использования метрик Холстеда при экспертной оценке программного обеспечения информационно – управляющих систем критического применения. Получен универсальный алгоритм расчёта метрик Холстеда для ассемблерных программ. На его основе разработано инструментальное средство (утилита) поддержки расчета метрик Холстеда. Утилита была апробирована на программном обеспечении, написанном на языке ассемблер ICC430 для компьютерной системы аварийной защиты ядерного реактора ВВЭР-1000

программное обеспечение, метрики Холстеда, экспертная оценка, утилита, информационно-управляющая система, алгоритм, инструментальное средство

Введение

Применение компьютерных информационных и управляющих систем (ИУС) в критических отраслях привело к тому, что такие системы и их компоненты (в первую очередь, программное обеспечение) все больше влияют на безопасность критических объектов. Одним из методов обеспечения безопасности критических объектов является государственное лицензирование в области критических технологий, которое проводится государственными регулирующими органами на основе нормативных документов (НД). Стандарты и отраслевые НД содержат требования по безопасности к ИУС и их компонентам: техническим средствам (ТС) и программному обеспечению (ПО). При оценке ИУС, ТС и ПО на соответствие требованиям по безопасности возникает проблема поиска объективных критериев, на основании которых может быть осуществлена подобная оценка. Одним из таких методов является метрический подход к оценке качества, надежности и безопасности ПО, когда каждому из вербальных требований ставится в соответствие множество метрик, предназначенных для объективной количественной оценки выполнения данного требования.

Одними из универсальных метрик являются метрики Холстеда, разработанные в 1970-х г.г. и утвержденные в качестве стандартизованных для оценки надежности ПО стандартом IEEE 982.1-1988 «IEEE Standard Dictionary of Measures to Produce Reliable Software». Поэтому, представляется целесообразным применить метрики Холстеда для оценки соответствия ПО требованиям к структуре и объему программных модулей. Подобное требование к ПО в той или иной интерпретации содержится в имеющихся НД в таких критических отраслях как атомная энергетика, авиация, космонавтика.

Метрики Холстеда и опыт их применения описаны в работах [1-5]. Основы метрического подхода к оценке качества ПО проанализированы в работах [6,7]. Анализ влияния КСУ и ПО на безопасность критических приложений проведен в работах [8-10]. Однако в известных публикациях отсутствует анализ возможности использования тех или иных метрик при оценке безопасности критического ПО.

Целью данной статьи является анализ применимости и разработка элементов методики использования метрик Холстеда при оценке безопасности ПО критических ИУС.

Для этого решены следующие задачи:

– на основе анализа метрик Холстеда разра-

ботан алгоритм для определения входных значений (примитивов) для расчета метрик;

– разработано инструментальное средство (ИС) для статического анализа ПО;

– на конкретном примере для ПО критического применения проведено определение метрик Холстеда и проиллюстрировано их использование при анализе безопасности ПО;

– разработана методика оценки качества программ, написанных на языке Assembler, с использованием метрик Холстеда.

1. Описание метрик Холстеда

Исходными данными для расчета метрик являются [1]:

– число различающихся простых операторов (словарь операторов) $n1$;

– число различающихся простых операндов (словарь операндов) $n2$;

– общее число всех операторов $N1$;

– общее число всех операндов $N2$;

– число различных входных и выходных параметров $n2^*$.

Основными метриками Холстеда являются:

1) словарь: $n = n1 + n2$;

2) длина программы: $N = N1 + N2$;

3) уравнение (оценочное) длины программы:

$\hat{N} = n1\log_2 n1 + n2\log_2 n2$;

4) объем программы: $V = N\log_2 n$;

5) потенциальный объем программы:

$V^* = (2 + n2^*)\log_2(2 + n2^*)$;

6) сложность программы: $D = (n1/2)(N_2 / n2)$;

7) уровень программы: $L = 1/D$;

8) усилия на разработку программы: $E = V/L$;

9) количество ошибок в программе:

$B = V/3000 = E^{2/3}/3000$;

10) время разработки программы: $T = E/S$,

где S = 18 – число Страунда (количество элементарных мыслительных операций в секунду).

2. Алгоритм и инструментальное средство для расчета метрик Холстеда

Предлагается универсальный алгоритм расчёта метрик Холстеда для программ, написанных на языке программирования (ЯП) Assembler. Универсальность алгоритма обеспечивается созданием базы данных, в которой хранятся различные системы команд. При анализе различных видов Assembler из базы данных загружается массив, содержащий соответствующую систему команд, или же создается новый массив, если данная разновидность Assembler ранее не анализировалась. Аналогичным образом может быть проанализирован любой другой ЯП. Алгоритм расчёта метрик Холстеда состоит из семи этапов, которые представлены на рис. 1.

Универсальный алгоритм расчёта метрик Холстеда является основой для разработки инструментального средства (ИС). Разработанная утилита производит расчет метрик Холстеда, в том числе и значение количества ошибок в ПО. Утилита была разработана на языке PHP в качестве Web-приложения, и предназначена для поддержки экспертизы ПО критического применения, написанного на ЯП Assembler.

Окно утилиты расчета метрик Холстеда представлено на рис. 2. Для определения значений метрик необходимо выполнить следующую последовательность действий:

– загрузить интерфейсное окно утилиты через глобальную сеть Интеренет или на локальном сервере посредством любого Web-браузера (Internet Explorer, Opera, Netscape Navigator);

– при помощи кнопки “Обзор...” выбрать файл с программой на языке Assembler ICC430 для расчёта метрик Холстеда;

– запустить утилиту кнопкой “Далее”.

Вычисление метрик Холстеда осуществляется автоматически и результат выводится в интерфейсном окне. Кроме значений метрик, в окне утилиты даётся перечень операторов, используемых в

Assembler ICC430, и количество операторов по типам в анализируемом файле.

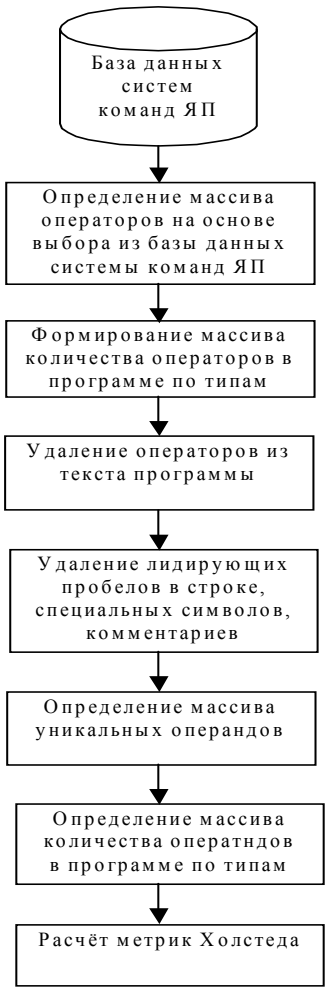

Рис. 1. Алгоритм расчёта метрик Холстеда

3. Определение метрик Холстеда для ПО системы аварийной защиты

Апробация предложенного ИС было выполнено для ПО программно-технического комплекса аварийной и предупредительной защиты ядерного реактора (ПТК АЗ-ПЗ). Данный ПТК был разработан специалистами ЗАО «Радий» (г. Кировоград) для модернизации морально и физически устаревших аналогичных систем АЭС Украины. ПТК АЗ-ПЗ обладает следующими особенностями, обусловленными спецификой объекта управления и требованиями к ИУС по ЯРБ:

– наличие двух диверсных комплектов, каждый из которых состоит из трех резервированных каналов и решающего мажоритарного устройства, построенного по сетевой схеме;

– архитектура ПТК включает два уровня:

верхний (АРМ оператора) и нижний (распределенная микропроцессорная сеть);

- ПО нижнего уровня написано на ЯП Assembler ICC430; ПО верхнего уровня написано на ЯП С++;ПО нижнего уровня представляет собой распределенную иерархическую структуру «ПО канала – ПО блоков – ПО процессоров – программные модули»;процессоры нижнего уровня выполняют функции обработки сигналов от датчиков, диагностики и контроля; управляющие функции реализованы при помощи ПЛИС.

В табл. 1 показана часть структуры ПО блоков, непосредственно выполняющих функции безопасности. Блоки БВА, БВД, БВТ обрабатывают входные сигналы от датчиков, блок БФЗ формирует сигналы защиты посредством программируемой логической структуры на ПЛИС и функционального процессора, реализующего управляющий алгоритм «преобразование давления в температуру». Также представлены результаты оценки модулей ПО ПТК АЗ-ПЗ по Холстеду, учтенные при тестировании.

4. Методика оценки ПО с использованием метрик Холстеда

Регулирующие требования Украины к ПО ИУС АЭС содержатся в НП 306.5.02/3.035-2000 «Требования по ядерной и радиационной безопасности к информационным и управляющим системам, важным для безопасности атомных станций». Согласно этому НД, одно из требований к ПО формулируется так: «ПО должно иметь модульную структуру. Текст одного модуля должен содержать ограниченное количество операторов, иметь ясную структуру, быть легко модифицируемым и тестируемым». Но данный НД не содержит критерия для определения объема, при соответствии которому программные модули отвечают указанным требованиям.

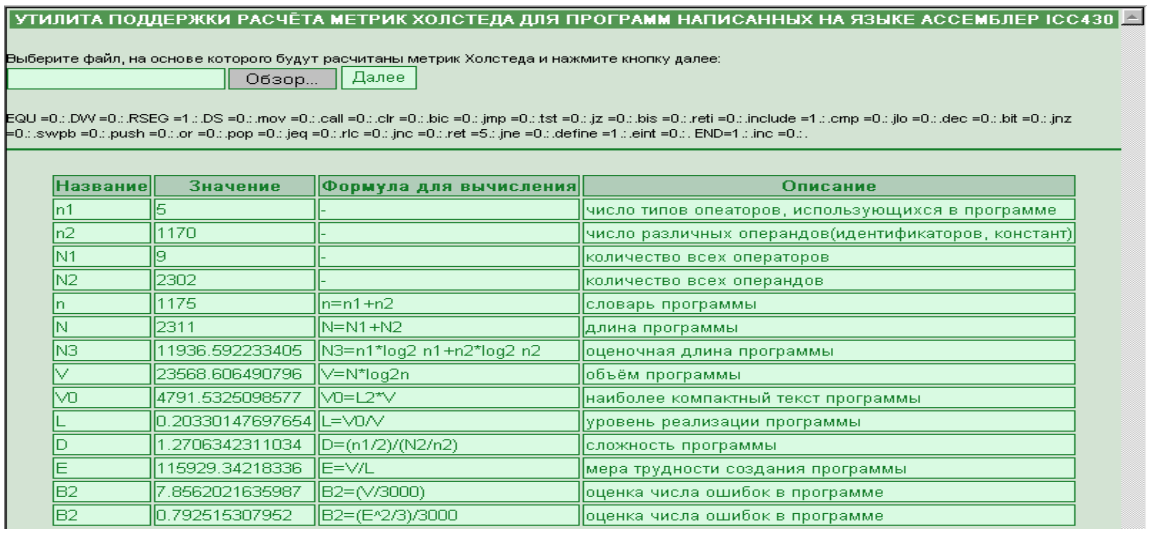

Рис. 2. Интерфейсное окно утилиты поддержки расчёта метрик Холстеда

Таблица 1

Структура ПО ПТК АЗ-ПЗ и вычисленные значения метрик Холстеда

Тип блока	Название процессора	Название ПО процессора	Назначение ПО процессора	Программные модули	Значение метрики, *B*	Ожидаемое количество ошибок, *[B]*
БВА	Функциональный	ПО ФП БВА	Прием и обработка сигналов от АЦП и передача их в КП	main.s43	0,34	0
				bva.s43	1,79	1
	Коммуникационный	ПО КП БВА	Прием и буферизация сигналов от ФП и передача их в БФС	main.s43	1,13	1
	Диагностический	ПО ДП БВА	Диагностика БВА	main.s43	1,40	1
БВД	Функциональный	ПО ФП БВД	Прием и обработка сигналов от дискретных входов и передача их в КП	main.s43 соответствует ФП БВА	0,34	0
				bvd.s43	0,45	0
	Коммуникационный	ПО КП БВД	Прием и буферизация сигналов от ФП и передача их в БФС	main.s43 соответствует КП БВА	1,13	1
	Диагностический	ПО ДП БВД	Диагностика БВД	main.s43 соответствует ДП БВА	1,40	1
БВТ	Функциональный	ПО ФП БВТ	Прием и обработка сигналов от АЦП и передача их в КП	main.s43	0,21	0
				adc.s43	0,07	0
				filters.s43	0,09	0
				timers.s43	0,01	0
				trans.s43	0,74	0
				uart1.s43	0,06	0
	Коммуникационный	ПО КП БВТ	Прием и буферизация сигналов от ФП и передача их в БФС	main.s43 соответствует КП БВА	1,13	1
	Диагностический	ПО ДП БВТ	Диагностика БВД	main.s43 соответствует ДП БВА	1,40	1
БФЗ	Функциональный	ПО ФП БФЗ	Преобразование значения давления в значение температуры	main.s43	0,16	0
				ptot.s43	0,79	0
				timers.s43	0,02	0

Расчёт метрик Холстеда позволяет оценить количественную сложность программы, которая определяется на основе полученных значений параметров. Итогом расчета метрик является показатель

количества дефектов в программном модуле B. На наш взгляд, данную метрику можно использовать для дополнительной оценки выполнения требования к модульной структуре ПО.

Если для программного модуля $B < 1$, то вероятность отсутствия дефектов в программе близка к единице. Чем B ближе к нулю, тем вероятность отсутствия дефектов выше. Данный факт также может служить подтверждением того, что декомпозиция ПО на модули была произведена рационально.

При $B \geq 1$ вероятность наличия дефектов в программе близка к единице. В этом случае должны быть дополнительно рассмотрены результаты тестирования ПО, так как метрика Холстеда оценивает количество дефектов непосредственно после завершения кодирования, без учета дефектов, устраняемых в процессе тестирования.

Более детальное представление элементов методики приведено на рис. 3.

Изложенное выше позволяет разработать методику оценки соответствия ПО указанным выше регулирующим требованиям с использованием метрик Холстеда. Методика может применяться как разработчиками ПО, так и экспертами, выполняющими анализ его безопасности.

Для применения указанной методики разработчику необходимо:

– декомпозировать программные модули таким образом, чтобы значение метрики Холстеда соответствовало $B < 1$;

– провести тестирование декомпозированных модулей. В случае, если декомпозиция не проводилась, для модулей, у которых $B \geq 1$, необходимо провести дополнительное тестирование, подтверждающее отсутствие дефектов;

– предоставить для экспертизы информацию, относящуюся к декомпозиции модулей, отчеты о тестировании после декомпозиции, а также отчеты о выявленных дефектах.

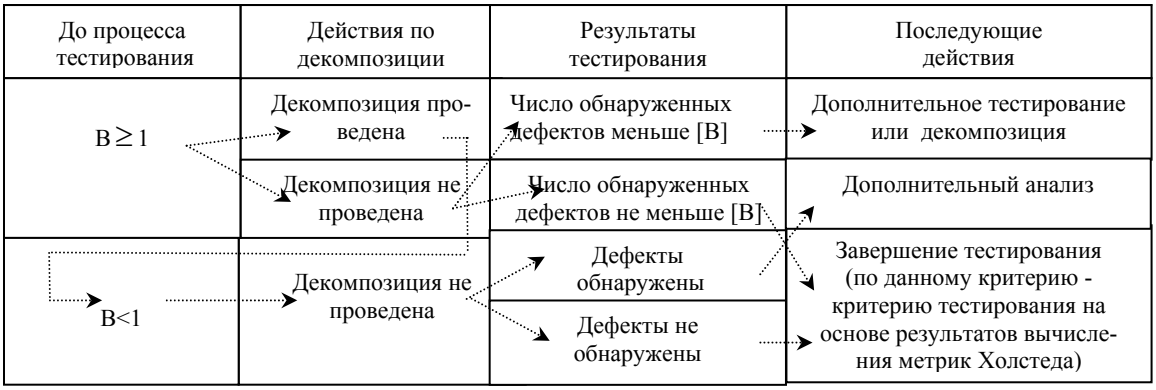

Рис. 3. Элементы методики

Эксперты, выполняющие анализ безопасности ПО, должны:

– проанализировать предоставленные отчеты на предмет корректности действий разработчика;

– оценить декомпозированные модули на основе метрик Холстеда (с использованием инструментального средства);

– составить заключение, отражающее соответствие ПО требованиям по безопасности в части требований к модульной структуре;

– при необходимости обосновать целесообразность проведения дальнейшей декомпозиции модулей и тестирования.

Для ПО ПТК АЗ-ПЗ применение предлагаемой методики позволило сделать следующие выводы.

По окончанию автономного тестирования для всех программных модулей оставшееся количество дефектов $B < 1$ (табл. 1), т.е. ПО ПТК АЗ-ПЗ соот-

ветствует предъявляемым требованиям по безопасности в части требований к модульной структуре.

Субъектная диверсность в ходе разработки ПО ПТК АЗ-ПЗ обеспечивалась таким образом. ПО БВА и БВД разрабатывалось одним программистом, а ПО ФП БВТ – другим, причем для КП БВТ и ДП БВТ использовалось ПО, разработанное первым программистом для БВА и БВД. Применение метрик Холстеда позволило сделать вывод, что для ПО ФП БВТ декомпозиция на модули является более удачной, поскольку для всех программных модулей оцененное количество дефектов B оказалось меньше. Таким образом, с точки зрения безопасности лучшими характеристиками обладает ПО ФП БВТ.

Заключение

Применение метрик Холстеда позволило повысить достоверность оценки выполнения регулирующих требований к ПО по ядерной и радиационной безопасности в части требований к модульной структуре. Для определения метрик был разработан универсальный алгоритм, который не зависит от применяемого ЯП. На основании алгоритма было разработано ИС для статического анализа ПО. Представляется целесообразным проведение анализа применимости других метрик для оценки безопасности критического ПО.

Литература

1. Холстед М.Х. Начала науки о программах.– М.: Финансы и статистика, 1981.– 128 с.
2. Кривченков А.А. Особенности применения метрики программного обеспечения при программировании микропроцессоров на Ассемблере // Программирование.– 1988.– № 5.– С. 46-55.
3. Шабалин А.Н. О росте сложности разрабатываемой программы // Программирование.– 1988.– № 6.– С. 23-28.
4. Шабалин А.Н., Дейков А.И. Проверка уравнения длины для Бейсик-программы // Программирование.– 1988.– № 2.– С. 86-89.
5. Апостолова Н.А., Гольдштейн Б.С., Зайдман Р.А. О программометрическом подходе к оценкам программного обеспечения // Программирование.– 1995.– № 4.– С. 38-44.
6. Харченко В.С., Тарасюк О.М., Скляр В.В. О метрическом подходе к оценке качества и надежности программного обеспечения // Системи обробки інформації.– Харків: НАНУ, ПАНМ, ХВУ.– 2002.– Вип. 6 (22).– С. 342-345.
7. Huang S.-J., Lai R. Measuring the Maintainability of a Communication Protocol Based on Its Formal Specification // IEEE Transactions on Software Engineering.– 2003.– Vol. 29.– n4.– P. 327-344.
8. Ястребенецкий М.А., Розен Ю.В., Виноградская С.В., Васильченко В.Н. и др. Нормирование и оценка безопасности информационных и управляющих систем. Цикл статей (1-9) // Ядерная и радиационная безопасность.– 2001-2002.
9. Конорев Б.М., Харченко В.С., Чертков Г.Н. Концепция и принципы реализации интегрированной инструментальной системы для поддержки экспертизы и независимой верификации критического программного обеспечения.– Харьков: Сертцентр АСУ, 2003.– 60 с.
10. Харченко В.С., Ястребенецкий М.А., Скляр В.В. Новые информационные технологии и безопасность информационно-управляющих систем АЭС // Ядерная и радиационная безопасность.- 2003.– Т. 6.– № 2.– С. 19-28.
11. Харченко В.С., Скляр В.В., Ястребенецкий М.А. Экспертная оценка безопасности OTS компонент информационных и управляющих систем АЭС // Збірник наукових праць Інституту проблем моделювання в енергетиці. Спеціальний випуск "Інформаційні технології в енергетиці".– К.: ІПМЕ. – 2003.– С. 12–19.

Поступила в редакцию 11.10.03

Рецензент: д-р техн. наук, профессор Сироджа И.Б., Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», г. Харьков