УДК 629.7.05+004.05

Ю.С. МАНЖОС

Національний аерокосмічний університет «ХАІ», Україна

ВИКОРИСТАННЯ АНАЛІЗУ РОЗМІРНОСТЕЙ ДЛЯ ПІДВИЩЕННЯ РІВНЯ ФУНКЦІОНАЛЬНОЇ БЕЗПЕКИ I&C СИСТЕМ

Розглянуто основи інформаційної технології підвищення рівня функціональної безпеки, побудованої на принципах аналізу фізичних розмінностей, що дозволяє верифікувати програмний код під час його розробки, зменшити ризики некоректного використання вхідних даних, та контролювати відсутність функціональних відмов під час штатної роботи технічних систем. Наведено перелік операцій, що контролюються. Визначені межі контролю коду під час формальної верифікації, а також контролю обчислювальних процесів під час експлуатації систем. Приведені класи програмних дефектів, та визначена ефективність методу, що для реального коду перевищує 80%.

Ключові слова: *аналіз розмірностей, статичний аналіз, формальна верифікація, функціональна безпека*

Вступ

Безпека життя сучасної людини неможлива без функціонування I&C систем. Тому ймовірність їх функціональних відмов має бути надзвичано низькою. Сучасна нормативна база, наприклад стандарт ISO/IEC 61508 [1, 2], наділяє велику увагу методам забезпечення функціональної безпеки (ФБ).

В умовах глобалізації світової економіки одним з важливих факторів ризику є ризики, обумовлені неузгодженістю обраних для розробки складних технічних систем одиниць виміру, наприклад SI, CGS. та несистемних одиниць, кількість яких вимірюється сотнями. Під час міжнародної кооперації у розробці програмного забезпечення (ПЗ) важливим є коректне застосування десятинних префіксів фізичних величин.

Забезпечення ФБ вимагає особливих потреб до ПЗ I&C систем, та проявів програмних дефектів (ПД). Рівень функціональної безпеки може не забезпечуватися через ПД, пов'язані з використанням некоректних адрес програмних даних, наприклад вихід за межі масивів. Небезпечним є також прояв алгоритмічних помилок, помилок кодування, обумовлених використанням некоректних операцій та операндів.

Тому виникає потреба у створенні спеціальних методів, що зменшуть ризики функціональних відмов I&C систем, обумовлених застосуванням різних систем одиниць, позасистемних одиниць, розробниками-учасниками міжнародних кооперацій, а також звичайними помилками проектування та кодування ПЗ.

Одним із підходів є використання математичного апарату аналізу розмінностей [3], що дозволяє ідентифікувати значну кількість ПД у ПЗ I&C систем під час їх розробки. Крім того, використання методів аналізу розмірностей під час функціонування технічних систем значно підвищіть ФБ завдяки додатковому контролю достовірності обчислюваних процесів. Відомо кілька методів, заснованих на аналізі розмінностей [4]. Один із варіантів такого методу [5] - семантичний контроль має певні недоліки. обмеженість однією системою одиниць; неможливість використання позасистемних одиниць десятинних префіксів; цілочисельне обмеження степені, необхідність задання розмірностей для всіх програмних змінних. Для подолання вказаних недоліків і було розроблено запропонований далі метод.

1. Основна ідея методу

Відомо[1, 2], що стандарт 61508 містить певні вимоги до ПЗ, що використовується у I&C системах. Наприклад необхідність контролю розмірностей вхідної інформації, областей памяті та періодичних перевірок відсутності функціональних відмов у каналах керування.

У той же час наявність кількох систем одиниць, кожна з яких визначає кілька сот фізичних величин, існування системи десятинних префіксів, що мають як міжнародні так і національні позначення та найменування, обумовлює певні ризики під час кооперації міжнародних колективів.

Аналіз розмірностей, визначає, що будь-яка з фізичних величин може бути подана як добуток деяких степеней кількох базових величин. Найпоширеніша з систем одиниць SI визначає як базові: довжину, час, масу, термодинамічну температуру, електричний струм, яскравість та кількість речовини, а

всі одиниці виміру фізичних величин визначає як добуток степенів базових одиниць виміру.

Система SI має такі базові одиниці виміру: метр, секунда, кілограм, градус Кельвіна, Ампер, кандела, моль. Всі похідні одиниці є добутками відповідних степеней, наприклад Ньютон=кілограм * метр /секунда^2, або кг^1 м^1 сек^2.

Використання степеней базових одиниць як систему координат дозволяє розглядати так званий семантичний простір (що є метричним та лінійним) та верифікувати програмний код на підставі головного постулату аналізу розмірностей: всі елементи алгебраїчних сум, порівнянь тощо, мають бути узгодженими, тобто мати еквівалентні фізичні розмірності. У нашому разі вектори, утворені зі степенем базових одиниць мають збігатися. Для підвищення ефективності семантичної верифікації слід доповнити множину базових одиниць радіаном та стерадіаном. Це надасть можливість контролювати обчислення, пов'язані з використанням тригонометричних функцій: аргументи прямих тригонометричних функцій мають бути у радіанах а результат безрозмірним, тобто мати нульовий вектор розмірності. У той же час arc- функції навпаки повинні мати аргумент безрозмірним , а результат у радіанах. У випадку експоненційних та логарифмічних функцій як аргумент так і результат мають бути безрозмірними.

Таким чином, кожна з алгебраїчних операцій може бути проконтрольована з точки зору аналізу розмірностей. Слід зазначити, що у випадку незадання початкової фізичної розмірності для проміжних програмних змінних коректність відповідних виразів може бути також проконтрольована на предмет порядку обчислень. Коректність алгебраїчних операцій та розмірність результату приведено у таблиці 1.

Таблиця 1

Коректність операцій та розмірність результату

Розмірність операндів		Операції										
$\vec{a}$	$\vec{b}$	$a = b$		$a \pm b$		$a \times b$		a/b		a^b		$a \approx b$
		c	$\vec{r}$	c	$\vec{r}$	c	$\vec{r}$	c	$\vec{r}$	c	$\vec{r}$	c
0	0	+	0	+	0	+	0	+	0	+	0	+
0	u	-	0	-	0	-	u	-	u	-	0	-
0	$\vec{b} \neq 0$	-	0	-	0	+	$\vec{b}$	+	$-\vec{b}$	-	0	-
u	0	+	0	-	u	-	u	-	u	-	u	-
u	u	-	u	-	u	-	u	-	u	-	u	-
u	$\vec{b} \neq 0$	+	$\vec{b}$	-	u	-	u	-	u	-	u	-
$\vec{a} \neq 0$	0	-	$\vec{a}$	-	$\vec{a}$	+	$\vec{a}$	+	$\vec{a}$	+	$\lvert b \rvert \vec{a}$	-
$\vec{a} \neq 0$	u	-	$\vec{a}$	-	$\vec{a}$	-	u	-	u	-	$\vec{a}$	-
$\vec{a} \neq 0$	$\vec{b} \neq \vec{a}$	-	$\vec{a}$	-	$\vec{a}$	+	$\vec{b}+\vec{a}$	+	$\vec{a}-\vec{b}$	-	$\vec{a}$	-
$\vec{a} \neq 0$	$\vec{b} = \vec{a}$	+	$\vec{a}$	+	$\vec{a}$	+	$\vec{b}+\vec{a}$	+	0	-	$\vec{a}$	+

В табл. 1 прийняті такі позначення:

$a \approx b$ операція порівняння; $\vec{a}, \vec{b}, \vec{r}$ – фізичні розмірності операндів та результату; $\lvert b \rvert$ – чисельна у звичайному сенсі складова програмної змінної; 0 – безрозмірна програмна змінна (всі складові вектору розмірності дорівнюють 0); u – невизначена розмірність програмної змінної;С- ознака семантичної коректності операції; «+» операція коректна; «-» операція некоректна

Використання методів аналізу розмірностей для контролю чинності системи може потребувати виконання операцій над нецілими числами – степенями базових одиниць виміру, що може мати місце, наприклад, внаслідок оптимізації порядку обчислень компілятором. Найпростіше рішення полягає в використанні для запису кожної з координат у семантичному просторі двох цілих чисел. Такий підхід дозволить використовувати системи з нецілими показниками степеней, наприклад CGS у якій одиниця електричного заряду визначається як грам$^{1/2}$ см$^{3/2}$ сек$^{-1}$. Для забезпечення контролю з раціональними степенями досить одного байту для чисельника та знаменника. Взагалі кожна одиниця даних потребуватиме додатково 18 байт. Але ця інформаційна надлишковість надасть ПЗ нову якість – можливість самоконтролю, а всій технічній системі – додаткову функціональну безпеку.

Подальше підвищення ФБ досягається за рахунок контролю як систем одиниць, так і десятинних префіксів. Відповідна модель даних зображена на рис. 1. Вона містить истеми одиниць, базові одиниці, мови розробників та користувачів технічної системи. Розмірність всіх одиниць (як похідних так і позасистемних). А також дані про чисельну залежність позасистемних та системних одиниць. Крім

того зберігається локальна інформація про використані одиниці виміру фізичних величин, а також десятинні префікси (назву та позначення). Кожна з програмних змінних має посилання на використану одиницю виміру та десятинний префікс. Це дозволяє мінімізувати ризики, пов'язані з некоректним вхідними даними.

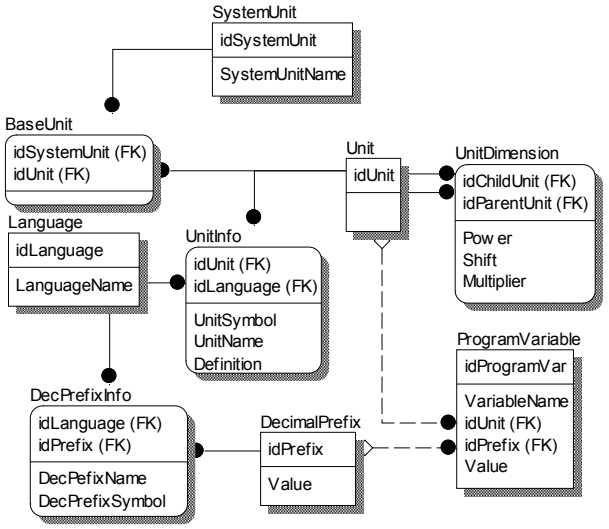

Рис. 1. Модель «Сутність-звязок» для програмних змінних

Розглянуті модель даних та таблицю коректності операцій було використано для розробки мовою C++ бібліотеки класів та шаблонів, що дозволило, за допомогою технології узагальненого програмування замінити стандартні програмні типи даних на відповідні шаблони та верифікувати формальним методом програмний код однієї з систем реального часу. Крім того, бібліотека дозволила автоматично трансформувати позасистемні одиниці до обраної системи одиниць та формувати результати обчислень у будь-якій іншій системі одиниць. При цьому всі ПД, позначені у таблиці 1 «-», а також некоректні виклики програмних функцій формували відповідні виключення та запис у log-файл. Враховуючи те, що діагностуюча здатність методу сягала адитивної операції, це значно зменшило трудомісткість тестування.

2. Аналіз ефективності методу

Визначимо ефективність методу як:

$$\eta = \frac{P_F}{P_F + P_N}, \qquad (1)$$

де P_F, P_N – ймовірності виявлення та пропуску ПД за умовами їх існування.

Для знаходження цих величин розглянемо зображену на рис. 2 ймовірнісну модель семантичних дефектів програмного забезпечення (ПЗ).

Будь-який елемент програми (вузол U) з ймовірностями ε_O, $\varepsilon_C = 1 - \varepsilon_O$ можуть бути відповідно операндам або операцією (командою) - вузли O, C. З ймовірностями ε_{EO}, $1 - \varepsilon_{EO}$ операнд може бути дефектним, або коректним (вузли E_O та Ok). Коректність операнду, що може бути як програмною змінною, так і константою, визначається коректним застосуванням з точки зору розмірності. Збіг фізичних розмірностей програмних змінних унеможливлює розпізнавання ПД (вузол N). Позначимо ймовірність збігу та незбігу розмінностей як ε_T, $1 - \varepsilon_T$. Розпізнання ПД позначено F. Вузли F, N, Ok, мають безумовний зв'язок з U, звідки з ймовірністю ε_C маємо перехід до C. Надалі будемо вважати, що всі операції поділено на: операції переходу, адитивні, мультиплікативні та виклики функцій з відповідними ймовірностями $\varepsilon_J, \varepsilon_A, \varepsilon_M, \varepsilon_F$ та вузлами C_J, C_A, C_M, C_F До адитивних віднесемо додавання, віднімання, порівняння та присвоєння. До мультиплікативних – множення, ділення та піднесення до степеню.

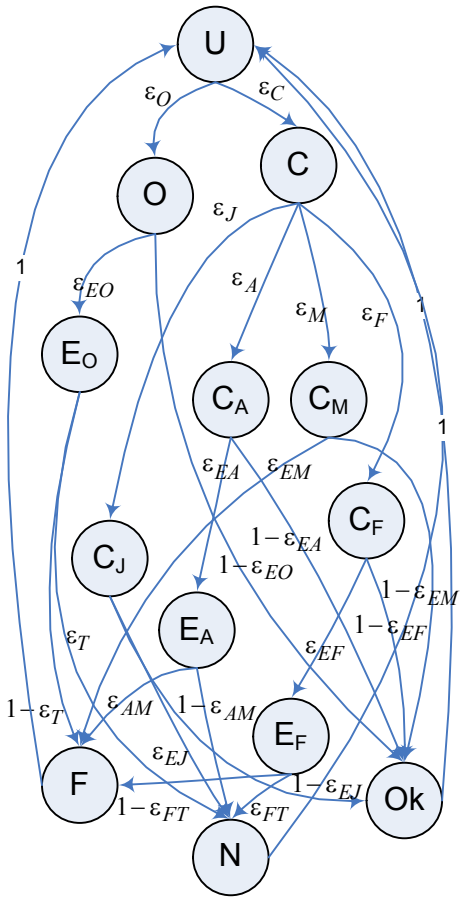

Рис. 2. Ймовірнісна модель семантичних ПД

Для операцій переходу з ймовірностями ε_{EJ}, $1-\varepsilon_{EJ}$ можлива альтернатива «дефект існує», «дефект відсутній. Існування ПД, пов'язаного з невірним переходом, не може бути розпізнано аналізом фізичної розмірності. Тому з ймовірністю ε_{EJ} має перехід до N, а з $1-\varepsilon_{EJ}$ до Ok.

У випадку виклику функцій, з ймовірностями ε_{EF}, $1-\varepsilon_{EF}$ маємо існування та відсутність ПД. Таким чином, з ймовірністю ε_{EF} переходимо до стану E_F «дефект виклику функції», а з $1-\varepsilon_{EF}$ – до стану Ok. Існування ПД, пов'язаного з невірним використанням викликів функцій, не може бути розпізнано аналізом фізичної розмірності тільки у випадку, коли фізичні розмірності всіх фактичних та формальних параметрів збігаються. Позначимо ймовірність такої події ε_{FT}. Тоді з ймовірністю ε_{FT} маємо перехід до N, а з $1-\varepsilon_{FT}$ до F.

Позначивши ймовірності існування та відсутності ПД адитивних операцій як ε_{EA}, $1-\varepsilon_{EA}$, маємо переходи до E_A та Ok. Існування ПД, пов'язаних з невірним використанням адитивних операцій, розпізнається аналізом фізичної розмірності тільки коли у результаті ПД має місце використання замість адитивної операції мультиплікативної. Позначимо ймовірність такої події ε_{AM}. Тобто з ймовірністю ε_{AM} маємо перехід до F, а з $1-\varepsilon_{AM}$ до N, що відповідає неможливості розпізнання ПД.

Мультиплікативні операції C_M, також можуть мати ПД, що обумовлює переходи до F, та Ok.

На підставі аналізу ймовірностей переходів маємо систему лінійних рівнянь (2).

$$\begin{cases} P_N + P_{OK} + P_F = P_U \\ P_U\varepsilon_C = P_C \\ P_C\varepsilon_M = P_{CM} \\ P_{CM}\varepsilon_{EM} + P_{EO}(1-\varepsilon_T) + \\ \quad +P_{EA}\varepsilon_{AM} + P_{EF}(1-\varepsilon_{FT}) = P_F \\ P_U\varepsilon_O = P_O \\ P_O(1-\varepsilon_{EO}) + P_{CM}(1-\varepsilon_{EM}) + P_{CA}(1-\varepsilon_{EA}) + \\ \quad +P_{CF}(1-\varepsilon_{EF}) + P_{CJ}(1-\varepsilon_{EJ}) = P_{OK} \\ P_O\varepsilon_{EO} = P_{EO} \\ P_C\varepsilon_A = P_{CA} \\ P_C\varepsilon_F = P_{CF} \\ P_{CA}\varepsilon_{EA} = P_{EA} \\ P_{CF}\varepsilon_{EF} = P_{EF} \\ P_C\varepsilon_J = P_{CJ} \\ P_{EA}(1-\varepsilon_{AM}) + P_{EF}\varepsilon_{FT} + \\ \quad +P_{EO}\varepsilon_T + P_{CJ}\varepsilon_{EJ} = P_N \end{cases} \quad (2)$$

Коректне розв'язання системи (2) для знаходження P_F, P_N вимагає урахування умови нормування:

$$P_U + P_C + P_{CM} + P_F + P_O + P_{OK} + P_{EO} + P_{OK} + \\ +P_{EO} + P_{CA} + P_{CF} + P_{CJ} + P_{EA} + P_{EF} + P_N = 1. \quad (3)$$

Заміна в (2) одного з рівнянь, наприклад рівняння (2.6) на (3) дозволяє отримати коректну систему (4).

$$\begin{cases} P_N + P_{OK} + P_F = P_U \\ P_U\varepsilon_C = P_C \\ P_C\varepsilon_M = P_{CM} \\ P_{CM}\varepsilon_{EM} + P_{EO}(1-\varepsilon_T) + \\ \quad +P_{EA}\varepsilon_{AM} + P_{EF}(1-\varepsilon_{FT}) = P_F \\ P_U\varepsilon_O = P_O \\ P_U + P_C + P_{CM} + P_F + P_O + \\ \quad +P_{OK} + P_{EO} + P_{CA} + P_{CF} + \\ \quad +P_{CJ} + P_{EA} + P_{EF} + P_N = 1 \\ P_O\varepsilon_{EO} = P_{EO} \\ P_C\varepsilon_A = P_{CA} \\ P_C\varepsilon_F = P_{CF} \\ P_{CA}\varepsilon_{EA} = P_{EA} \\ P_{CF}\varepsilon_{EF} = P_{EF} \\ P_C\varepsilon_J = P_{CJ} \\ P_{EA}(1-\varepsilon_{AM}) + P_{EF}\varepsilon_{FT} + \\ \quad +P_{EO}\varepsilon_T + P_{CJ}\varepsilon_{EJ} = P_N \end{cases} \quad (4)$$

Розв'язання системи (4) дозволяє знайти необхідні ймовірності розпізнання та пропуску ПД як функцій статистичних характеристик ПЗ

$$\begin{cases} P_F = \dfrac{\varepsilon_C\varepsilon_M\varepsilon_{EM} + \varepsilon_O\varepsilon_{EO}(1-\varepsilon_T) + }{3+\varepsilon_C+\varepsilon_O\varepsilon_{EO}+\varepsilon_C\varepsilon_A\varepsilon_{EA}+\varepsilon_C\varepsilon_F\varepsilon_{EF}} \rightarrow \\ \qquad \rightarrow \dfrac{+\varepsilon_C\varepsilon_A\varepsilon_{EA}\varepsilon_{AM} + \varepsilon_C\varepsilon_F\varepsilon_{EF}(1-\varepsilon_{FT})}{3+\varepsilon_C+\varepsilon_O\varepsilon_{EO}+\varepsilon_C\varepsilon_A\varepsilon_{EA}+\varepsilon_C\varepsilon_F\varepsilon_{EF}} \\ P_N = \dfrac{\varepsilon_C\varepsilon_A\varepsilon_{EA}(1-\varepsilon_{AM}) + \varepsilon_C\varepsilon_F\varepsilon_{EF}\varepsilon_{FT} + }{3+\varepsilon_C+\varepsilon_O\varepsilon_{EO}+\varepsilon_C\varepsilon_A\varepsilon_{EA}+\varepsilon_C\varepsilon_F\varepsilon_{EF}} \rightarrow \\ \qquad \rightarrow \dfrac{+\varepsilon_O\varepsilon_{EO}\varepsilon_T + \varepsilon_C\varepsilon_J\varepsilon_{EJ}}{3+\varepsilon_C+\varepsilon_O\varepsilon_{EO}+\varepsilon_C\varepsilon_A\varepsilon_{EA}+\varepsilon_C\varepsilon_F\varepsilon_{EF}} \end{cases}$$

Позначивши ймовірність існування ПД як $\varepsilon_D = \varepsilon_O\varepsilon_{EO} + \varepsilon_C(\varepsilon_M\varepsilon_{EM} + \varepsilon_A\varepsilon_{EA} + \varepsilon_F\varepsilon_{EF} + \varepsilon_J\varepsilon_{EJ})$ та припустивши, їх рівномірний розподіл $\varepsilon_{EM} = \varepsilon_{EA} = \varepsilon_{EF} = \varepsilon_{EJ} = d$, маємо, що ефективність методу визначається як

$$\eta = \varepsilon_C\varepsilon_M + \varepsilon_O(1-\varepsilon_T) + \varepsilon_C\varepsilon_A\varepsilon_{AM} + \varepsilon_C\varepsilon_F(1-\varepsilon_{FT}).$$

З урахуванням $\varepsilon_O + \varepsilon_C = 1$, $\varepsilon_A + \varepsilon_M + \varepsilon_F + \varepsilon_J = 1$ та після перетворень маємо:

$$\eta = 1 - \varepsilon_T - \varepsilon_C\left(\varepsilon_F\varepsilon_{FT} + \frac{\varepsilon_A}{2} + \varepsilon_J - \varepsilon_T\right), \quad (5)$$

де ε_C − ймовірність (частка) операцій у коді, $\varepsilon_J, \varepsilon_F$, ε_A − ймовірності (частки) переходів, викликів функцій та адитивних операцій, що визначаються як:

$$\varepsilon_C = \frac{N_C}{N_C + N_O},$$

де N_C, N_O − загальна кількість операцій (команд) та операндів;

$$\varepsilon_J = \frac{N_J}{N_C + N_J},\ \varepsilon_F = \frac{N_{CF}}{N_C + N_{CF}},\ \varepsilon_A = \frac{N_A}{N_C + N_A},$$

де N_J − загальна кількість переходів; N_{CF} − загальна кількість викликів функцій; N_A − загальна кількість адитивних операцій (додавань, віднімань, порівнянь, присвоєнь); ε_T − умовна ймовірність збігу фізичних розмірностей для різних змінних:

$$\varepsilon_T = \sum_{i=1}^{n} (1 - P_{Ti}) P_{Ti},$$

де P_{Ti} − частка операндів, що має еквівалентні розмірності та визначається як

$$P_{Ti} = \frac{N_{Ti}}{N_O + N_{Ti}},$$

де N_{Ti} − кількість операндів, що мають відповідну фізичну розмірність, а n − загальна кількість використаних у програмі операндів;

ε_{FT} − ймовірність збігу фізичних розмінностей формальних параметрів для різних програмних функцій, що визначається як:

$$\varepsilon_{FT} = \sum_{i=1}^{n} (1 - P_{FTi}) P_{FTi},$$

де P_{FTi} − частка функцій, що мають еквівалентні розмірності формальних параметрів та визначається як;

$$P_{FTi} = \frac{N_{FTi}}{N_F + N_{FTi}},$$

де N_F, N_{FTi} − відповідно загальна кількість функцій та кількість функцій, що мають еквівалентні фізичні розмірності формальних параметрів, а n − загальна кількість використаних у програмі функцій.

Середня ефективність методу визначиться як гіпероб'єм для (5) у просторі з базисом $\varepsilon_T, \varepsilon_C, \varepsilon_J, \varepsilon_F, \varepsilon_{FT}, \varepsilon_A$, обмеженому межами, що визначають діапазон ймовірностей $[0,1]$.

$$m_\eta = \frac{\int\limits_V \left(1 - \varepsilon_T - \varepsilon_C\left(\varepsilon_F \varepsilon_{FT} + \frac{\varepsilon_A}{2} + \varepsilon_J - \varepsilon_T\right)\right) dv}{\int\limits_v dv}, \quad (6)$$

де інтеграл обчислюється у шестивимірному просторі, за умовою $0 \le \varepsilon_F + \varepsilon_A + \varepsilon_J \le 1$, а елемент об'єму $dv = d\varepsilon_T d\varepsilon_C d\varepsilon_J d\varepsilon_F d\varepsilon_{FT} d\varepsilon_A$

В результаті чисельного інтегрування (6) для $0 \le \varepsilon_F + \varepsilon_A + \varepsilon_J \le 1$ та $0 \le \varepsilon_C, \varepsilon_T, \varepsilon_{FT} \le 1$ маємо середню ефективність m_η=0,49. Результати інтегрування (6), обчислені для реального коду з $0 \le \varepsilon_F, \varepsilon_T, \varepsilon_{FT} \le 0{,}1$ m_η=0,87.

Таким чином, для ПЗ, характерного для інформаційно-керуючих систем семантичний контроль забезпечує ефективність більше ніж 80%. Тобто більше 80 відсотків семантичних дефектів − некоректних використань операндів, операцій, функцій може бути виявлено завдяки використанню аналізу фізичних розмінностей як під час статичного аналізу, так і під час штатного функціонування системи

Висновки

В роботі розглянуто використання аналізу розмірностей, як подальшого розвитку семантичного контролю, для зменшення ризиків функціональних відмов складних технічних систем. Розглянуті основи інформаційної технології, що дозволила зменшити ризики, пов'язані з некоректним використанням фізичних величин та числових даних як вхідної інформації складних систем, обумовлених використанням під час розробки кількох систем одиниць, або некоректним переводом позасистемних фізичних одиниць та десятинних префіксів. Визначені межі контролю коду під час формальної верифікації, а також контролю обчислювальних процесів під час експлуатації систем. Приведені класи програмних дефектів, та визначена ефективність методу, що для реального коду перевищує 80%.

Література

1. Смит, Д. Функциональная безопасность. Простое руководство по применению стандарта МЭК 61508 и связанных с ним стандартов [Текст] / Д. Смит, К. Симпсон. – М.: Издательский дом «Технологии», 2004. – 208 с.

2. IEC 61508-7. International standard Functional safety of electrical/electronic/programmable electronic safety-related systems [Text]. – Part 7: Overview of techniques and measures, 2000.

3. Szirtes, T. Applied dimensional analysis and modeling [Text] / T. Szirtes // Butterworth–Heinemann, 2007. - 853 p.

4. Ferson, Scott. Making Sure Computers Don't Add Apples and Oranges: Automated Checking and Correction for Dimensional Balance and Unit Conformance [Electronic resource] / Scott Ferson // In ACM SIGSAM Bulletin, Communications in computer algebra. – September 2002. – Vol. 36, No. 3, Issue 141. – 9 p. – Access mode: http://www.sigsam.org/bulletin/articles/141/eccad_ab.pdf. – 30.11.2011 г.

5. Инвариантно-ориентированная оценка качества программного обеспечения космических систем [Текст] / Б.М. Конгрев, Ю.С. Манжос, В.С. Харченко, В.В. Сергиенко, Г.Н. Чертков. – Х.: Национальный аэрокосмический университет «ХАИ», Харьков, 2009. – 288 с.

Надійшла до редакції 23.02.2012

Рецензент: д-р техн. наук, проф. А.В. Дрозд, Одесский национальный политехнический университет, Одесса, Украина.

ИСПОЛЬЗОВАНИЕ АНАЛИЗА РАЗМЕРНОСТЕЙ ДЛЯ ПОВЫШЕНИЯ ФУНКЦИОНАЛЬНОЙ БЕЗОПАСНОСТИ I&C СИСТЕМ

Ю.С. Манжос

Рассмотрен метод уменьшения функциональных рисков, основанный на анализе размерностей программных данных, который лег в основу информационной технологии, улучшающей функциональную безопасность сложных технических систем. Предложенный метод позволяет с точки зрения анализа размерностей проверять корректность использования систем единиц, производных и внесистемных единиц, а также корректность программного кода в процессе его разработки, посредством статического анализа. Кроме того предложенный метод позволяет в процессе эксплуатации посредством соответствующих тестов контролировать ход управляющих вычислительных процессов. Показаны классы обнаруживаемых программных дефектов, оценена средняя эффективность метода в обнаружении программных дефектов, превышающая 80%..

Ключевые слова: анализ размерностей, статический анализ, формальная верификация, функциональная безопасность.

USING DIMENSIONAL ANALYSIS FOR IMPROOVING OF I&C SYSTEM'S FUNCTIONAL SAFETY

Y.S. Manzhos

A method of functional risk reduction is proposed. The method is based on dimensional analysis I&C system software. The proposed method is a base of Information Technology for improving functional safety. The Method allows to check the correctness of used system unit, decimal prefix, derived physical unit during the elaboration of I&C system software via formal verification by static analysis. The proposed Method allows to check of the working I&C system.

Keywords: dimensional analysis, static analysis, formal verification, functional safety.

Манжос Юрій Семенович – канд. техн. наук, доц. каф. інженерії програмного забезпечення Національного аерокосмічного університету "Харківський авіаційний інститут", e-mail: manzhos@ukr.net.