УДК 004.05, 004.415.5

Ю.С. МАНЖОС

Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Україна

БАГАТОІНВАРІАНТНИЙ МЕТОД ПІДВИЩЕННЯ НАДІЙНОСТІ ПРОГРАМНИХ ЗАСОБІВ

З метою підвищення надійності програмно-технічних комплексів розглянуто метод, що забезпечує контроль інтервальної, точностної та семантичної коректності обчислювальних процесів, та має у по рівнянні з відомими меншу ресурсоємність та більшу діагностуючу спроможність виявлення залишкових дефектів.

диверсність, інваріант, надійність програмного забезпечення

Вступ

Безпека сучасного суспільства визначається функціональної безпекою програмно-технічних комплексів (ПТК) критичного застосування. Одноканальна архітектура не дозволяє забезпечити ймовірність безвідмовної роботи вище за 0.8 на протязі 2000 .. 200 годин [1]. У той же час термін експлуатації таких систем вимірюється роками, що потребує пошуку нових методів забезпечення надійності. Необхідний рівень надійності досягається завдяки використанню багатоверсійної надмірності [2], що потребує значних ресурсів: часу, апаратури, співробітників.

Постановка задачі

Пропонується обмежитися розробкою однієї версії програмного забезпечення (ПЗ), а диверсифікацію розробки здійснювати завдяки контролю кількох інваріантів програмного забезпечення (ІПЗ). ІПЗ є властивістю, що зберігається у часі та не залежить від вхідних даних, що надходять до програмного засобу, а порушення інваріантних властивостей свідчить про наявність залишкових дефектів, які надалі будемо називати інваріантними дефектами (ІД). Одночасний контроль збереження кількох інваріантів під час стендових випробувань, чи штатного функціонування дозволить підвищити надійність програмного продукту та безпеку функціонування ПТК.

Як додаткові версії пропонується використати семантичне, інтервальне та точностне відображення, що здійснюються програмним кодом відповідно над семантиками (фізичними розмірностями), інтервалами (межами в яких повинні знаходитися дані) та точностями (абсолютними та відносними точностями) даних. Це дозволить без розробки додаткових версій функціонального ПЗ, а лише завдяки контролю семантичної, інтервальної та точностної коректності, діагностувати обчислювальні процеси у реальному часі, спростити дистанційну модифікацію та супроводження ПЗ, і, як наслідок, підвищити надійність сучасних ПТК критичного застосування. Підвищення надійності досягатиметься завдяки контролю збереження інваріантів під час виконання всіх операцій, та обчисленню статистичних характеристик ПЗ, потрібних для визначення меж надійності ПЗ у разі невиявлення інваріантних порушень.

Семантичний контроль (СК) [3, 4], що використовує векторну форму фізичної розмірності, в системах реального часу потребує десятикратного збільшення обсягів оперативного запам’ятовуючого пристрою та кількості операцій. Недоліком СК є те, що він дозволяє при виконанні програми тільки фіксувати факт порушення семантичного інваріанту, що значно зменшує діагностуючу здібність. Необхідна розробка більш ефективного, з точки зору ре-

сурсів та діагностики, методу контроля семантичної коректності ПЗ. Реалізація інтервального та точностного контролю (ІК) [3,4] потребує попереднього визначення інтервалів для всіх, у тому числі і тимчасових програмних змінних, що не завжди можливо через необхідність розв'язання систем лінійних рівнянь та нерівностей.

Реалізація методу багатоінваріантного контролю

В основу багатоінваріантного контролю (БІК) покладено наступне:

1. Цілочислельне семантичне дескрипторне відображення (ЦСДВ) та дескрипторну алгебру, що дозволяють верифікувати коректність ПЗ завдяки контролю збереження семантичних інваріантів [3] під час операцій над програмними змінними. ЦСДВ проєцує семантичний простір (СП) [3] на цілочисельну вісь (одновимірний простір цілочисельних семантичних дескрипторів (ЦСД)) та зменшує не тільки обсяг додаткової пам'яті, але й кількість додаткових операцій, необхідних для реалізації семантичного контролю. Правила перетворення ЦСД під час виконання операцій та умови їх коректності визначаються дескрипторною алгеброю, що дозволяю контролювати коректність окремих арифметичних та логічних виразів.

2. Інтервальне відображення та інтервальну алгебру [3], що дозволяє верифікувати коректність ПЗ завдяки контролю збереження інтервальних інваріантів під час операцій над програмними змінними. Завдяки відомим інтервалам програмних змінних та інтервальній алгебрі, що визначає правила перетворення інтервалів під час виконання програмних операцій, можливо контролювати не тільки коректність арифметичних та логічних виразів, але і їх послідовність.

3. Точностне відображення та точностну алгебру [3], що дозволяє верифікувати коректність ПЗ завдяки контролю збереження точностних інваріантів під час операцій над програмними змінними. Завдяки відомій точності програмних змінних та точностній алгебрі, що визначає правила перетворення точностей під час виконання програмних операцій, можливо додатково контролювати не тільки коректність арифметичних та логічних виразів, але і їх послідовність.

Кожне з перелічених відображень базується на окремих вхідних даних і дозволяє розглядати ПЗ як сукупність трьох відповідних моделей, що дозволяє підвищити надійність методу.

Найпростіша реалізація інварінатного контролю (ІК) можлива алгоритмічними мовами, що дозволяють перевизначення операцій. Адитивні операції: *присвоєння, додавання, віднімання, порівняння, виклики функцій;* а також математичні функцій: *модуль, sin, cos, tg, arcsin* тощо, повинні додатково оцінювати коректність використання операндів та результатів обчислення з точки зору фізичної розмірності. При цьому ЦСД операндів програмних операцій та формально-фактичних параметрів мають збігатися, а ЦСД аргументів математичних функцій повинні мати певні значення, наприклад відповідати розмірності *радіан*. Інтервальний та точнісний контролі надають додаткові умови на коректність програмного коду. Мультиплікативні операції: *множення, ділення тощо* повинні формувати нові розмірності, точності, інтервали, коректність яких має перевірятися перевизначеними адитивними операціями. Інваріантні порушення мають генерувати виключні ситуації або сигнали у керуючий процес. Діагностування ІД (рис. 1) виконується з точністю до адитивної операції. Для полегшення визначення місць програмного коду, що мають ІД, паралельно з виконанням основної задачі та діагностування відновлюється програмний код. Для цього невизначені операції на підставі аргументів формують бінарне дерево арифметичного виразу. У разі виявлення ІД сформоване дерево переводиться у текстову форму та передається у відповідний потік виводу, який може бути поєднано як з окремим файлом, так і з віддаленим процесом.

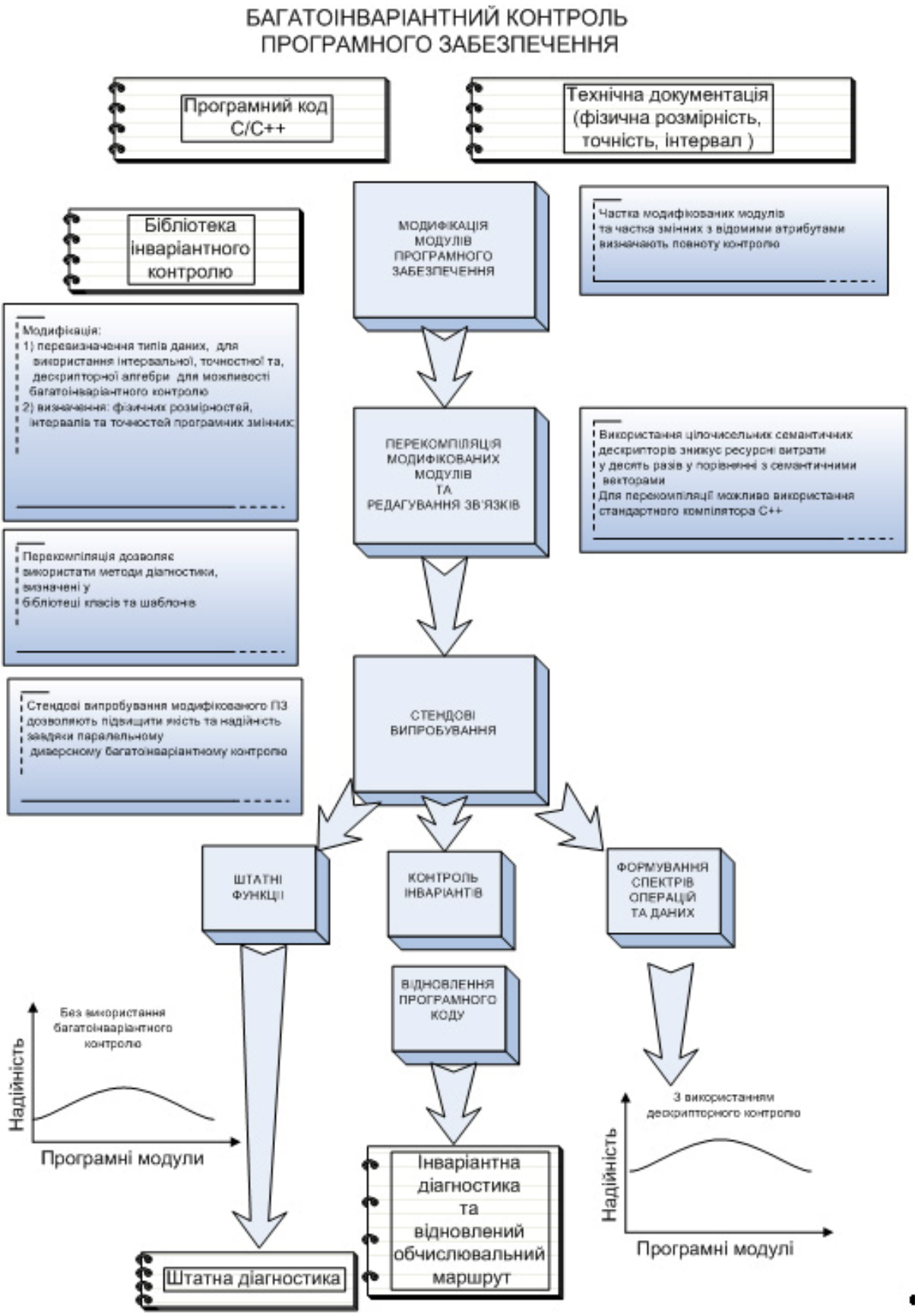

Рис. 1. Схема багато інваріантного контролю програмного забезпечення

Використання БІК під час стендових випробувань ПЗ дозволяє більш точно оцінити ймовірність відсутності ІД у коді, що верифікується, бо відомі моделі інтервальних, точностних та семантичних дефектів враховують лише статичний розподіл даних та операцій у програмному коду, у той же час ПЗ у різних режимах роботи використовує різні підмножини програмних модулів та даних, тому і ймовірності існування залишкових дефектів будуть для них різними. Перевизначення операцій дозволяє накопичувати інформацію про інтервальний, семантичний, точностний та операційний спектри та отримувати більш точне значення ймовірності існування залишкових ІД.

В зв'язку з тим, що у ПЗ використовуються програмні змінні різних типів даних, для перевизначення типів даних використовується шаблона класу, що перевизначає основні типи даних та множина шаблонів функцій, що перевизначають операції над даними. Шаблони класу та функцій оформлені як бібліотека, що реалізована стандартною мовою С++.

Висновки

Запропонований метод багатоінваріантного контролю, що забезпечує підвищення надійності програмних засобів, завдяки диверсифікації контроль інваріантної коректності, має у порівнянні з існуючими на порядок меншу потребу у ресурсах, високу діагностуючу спроможність та більш точну оцінку ймовірності існування залишкових інваріантних дефектів.

Застосування методу доцільно як під час стендових випробувань, що дозволить підняти надійність програмних засобів, такі у реальному часі, що підвищить рівень безпеки експлуатації інформаційно-керуючих систем для АЕС та авіа-космічних комплексів.

Подальші дослідження доцільно проводити у напрямку розширення множин програмних інваріантів, та програмних мов на яких реалізовані ПТК.

Література

1. Артеменко Е.А. Резервирование, диагностирование, восстановление – система обеспечения надежности сложных технических комплексов // Модели и системы. – 1999. – № 1. – С. 4-7.

2. Харченко В.С. Многоверсионные цифровые системы, важные для безопасности: анализ и перспективы // Модели и системы. – 1999. – № 1. – С. 61-64.

3. Конорев Б.М., Алексеев Ю.Г., Клименко Т.А., Манжос Ю.С, Петрик В.Л., Сергиенко В.В., Харченко В.С., Чертков В.С. / Калибровка чувствительности методов статического анализа, используемых для оценки качества и безопасности ПО ИУС АЭС // Междунар. симпозиум «Измерения, важные для безопасности в реакторах». – М.: Институт проблем управления им. Трапезникова, 2004. – С. 15-1 – 15-12.

4. Харченко В.С., Манжос Ю.С., Петрик В.Л. Статистический анализ программного обеспечения системы управления космическим аппаратом и оценка проверяющей способности семантического контроля // Технология приборостроения. – 2002. – № 2. – С. 52-59.

Надійшла до редакції 11.02.2008

Рецензент: д-р техн. наук, проф. Б.М, Конорев, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Харків.