

В связи с этим можно говорить о контроллере как о своеобразном "распределительном центре" (менеджере). Контроллер обрабатывает запросы пользователя, контроллер обращается к модели, контроллер же является посредником для вывода вида в браузер.

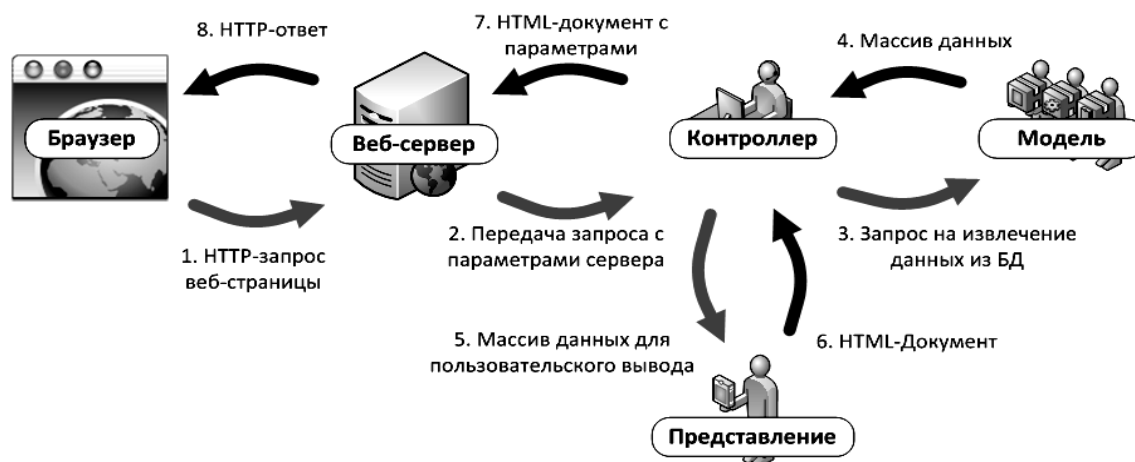


Рисунок 1 – Схема работы MVC для web-приложения

В рамках концепции MVC принято, что с базой должны работать только модели. Вывод: причины использовать MVC модель для разработки web-приложения:

- Типизированность. Каждый разработчик, который понимает MVC, быстро разберётся в чужом проекте (построенном по принципу MVC), и сможет продуктивно продолжить работу.
- Возможность повторного использования. Модель составляется так, чтобы предусмотреть многократное использование с минимальными изменениями или вообще без них.
- Расширяемость. Модель имеет четко определенное поле для работы, если требуется изменить вид или функциональность.

Список литературы

1. Интернет-ресурс: www.ozon.ru/context/detail/id/3771474/
2. Интернет-ресурс: www.rsdn.ru/article/patterns/generic-mvc.xml
3. Стивен Сандерсон ASP.NET MVC Framework с примерами на C# для профессионалов. — М.: Вильямс, 2009.
4. Интернет-ресурс: www.ibm.com/developerworks/ru/library/wi-embed/index.html?ca=dre-ru

УДК 004.05:004.42

О.М. Змеул

Науковий керівник – Доренський О.П., викладач
Кіровоградський національний технічний університет

Дослідження методів забезпечення надійності програмного забезпечення

Стрімкий розвиток інформаційних технологій (ІТ), комп’ютерної техніки, програмного забезпечення (ПЗ) та, як наслідок, технологій розробки програмних продуктів, розширення спектра використання автоматизованих систем у сучасному

світі зумовлюють паралельний розвиток усіх складових процесу побудови й впровадження ПЗ [1]. Підвищення складності та багатокomпонентності сучасного ПЗ вимагають спеціалізованого підходу під час його розробки й експлуатації. Будь-яке програмне забезпечення повинне досягати високого рівня надійності, що дозволяє йому ставати ефективним інструментом нових технологій. Будь-який збій ПЗ може мати серйозні наслідки та втрати (матеріальні та нематеріальні збитки), що є неприпустимим. Тому задача забезпечення надійності ПЗ, що розробляється, є актуальною та однією з основних для розробників та користувачів програмних продуктів.

За умови досягнення високої надійності, ПЗ стає ефективним та конкурентоспроможним. Саме від показника надійності похідними будуть інші, не менш важливі показники – якість, живучість, безпека, готовність [2].

Метою роботи є дослідження методів забезпечення надійності програмного забезпечення з метою уникнення збоїв в роботі ПЗ.

В багатьох дослідженнях поняття надійності програмного забезпечення (ПЗ) виділяють окремо [3], тому, що при застосуванні понять надійності до програмних засобів варто враховувати особливості і відмінності цих об'єктів від традиційних технічних систем, для яких спочатку розроблялася теорія надійності. Принципова відмінність програм від техніки, та технічних систем зокрема, полягає в тому, що програма не зношується з плином часу, а навпаки, виявляються помилки, які не були знайдені раніше, ПЗ з часом вдосконалюється і покращується. Водночас підвищуються і вимоги до надійності та витривалості програм, виникає потреба у скороченні затрат на тестування та, відповідно, у прогнозуванні надійності розроблюваного програмного забезпечення [2].

Аналіз [1-5] показав, що дослідження методів забезпечення надійності програмного забезпечення є актуальною задачею.

Визначальне значення для забезпечення надійності програм мають методи контролю і забезпечення надійності програм [5]. Для її підвищення і захисту інформації програмно-алгоритмічними методами ефективно застосовувати часову, інформаційну та програмну надлишковість.

Метод часової надлишковості полягає у використанні певної частини апаратних засобів і продуктивності комп'ютера для контролю виконання програм і відновлення обчислювального процесу (ОП). Передбачається певний резерв продуктивності, необхідний для контролю й підвищення надійності функціонування ПЗ. Резерв часу використовують для контролю і виявлення спотворень, їх діагностування і прийняття рішень щодо відновлення ОП або інформації, реалізації операцій відновлення.

Для застосування інформаційної надлишковості необхідно дублювати накопичені початкові й проміжні дані. Цей метод використовують для забезпечення достовірності даних, які потребують значного часу для відновлення чи найбільше впливають на нормальне функціонування програмного забезпечення.

Метод програмної надлишковості забезпечує контроль і достовірність найважливіших рішень з оброблення інформації та керування. При ньому застосовують кілька варіантів програм, що відрізняються методами розв'язання певного завдання або програмною реалізацією одного й того самого методу. Це дає змогу виключати спотворення результатів, зумовлених програмними помилками чи збоями ЕОМ.

Часто введенням надлишковості для забезпечення надійності ПЗ нехтують, що знижує показники надійності програм, що розробляють [5].

Щодо процесу експлуатації, то контроль надійності функціонування ПЗ проводять трьома етапами, відповідно до яких він і має назви.

- профілактичний контроль, за якого здійснюють у неробочому режимі системи керування або оброблення інформації під час проведення регламентних профілактичних робіт;
- передпусковий контроль, який проводять під час підготовки до ввімкнення нормального робочого режиму функціонування ПЗ;
- оперативний контроль, що здійснюють у процесі розв’язання системою основних функціональних завдань у нормальному режимі роботи.

Після завершення випробувань і передавання ПЗ замовнику його експлуатують і тиражують впродовж певного часу. На цьому етапі можливі спотворення програм у процесі їх зберігання та регулярної експлуатації. Крім того, з огляду на зміну вхідних даних, може виникнути необхідність внесення змін у програму. Тому навіть одне незначне низькоякісне коректування може спричинити істотне зниження напрацювання на відмову. Щоб зберегти і поліпшити показники надійності ПЗ під час його тривалої експлуатації, необхідно чітко регламентувати впровадження програм. Зміни в ПЗ вводять групами, що формує чергову версію ПЗ зі зміненими характеристиками.

Серед версій ПЗ виокремлюють еталонні і користувальницькі. Еталонну версію допрацьовують і модернізують основні розробники ПЗ або спеціалісти з експлуатації. Вони мають скоректовану технічну документацію, що відповідає програмам, і точний перелік усіх змін, введених у дану версію порівняно з попередньою. Розробка кожної наступної еталонної версії вимагає випробовування і перевірки на працездатність.

Користувальницькі версії (або версії конкретної системи [5]) формують за інструкціями і правилами, що є в експлуатаційній документації еталонних версій. Зміни в них обмежені й стосуються окремих компонентів ПЗ. Щоб перевірити, чи коректно їх виконують, додають методики перевірки та правила розроблення контролюючих тестів.

Загальними способами забезпечення і підвищення надійності ПЗ є:

- розроблення і вибір алгоритмів функціонування ПЗ, нечутливих до порушень обчислювального процесу (використання алгоритмічної надлишковості);
- удосконалення технології програмування;
- контроль і тестування програм з наступною їх корекцією;
- резервування програм та інші методи введення структурної надлишковості.

Отже, в роботі досліджено методи забезпечення надійності програмного забезпечення, запропоновано й обґрунтовано застосування часової, інформаційної й програмної надлишковості. Зважаючи на стрімкий розвиток ІТ-технологій, комп’ютерної техніки і програмного забезпечення, є перспективними подальші дослідження, зокрема у напрямку забезпечення надійності ПЗ мобільних систем.

Список літератури

1. Змеул О.М. Дослідження помилок програмного забезпечення / О.М. Змеул, О.П. Доренський // Актуальні задачі сучасних технологій: збірник тез доповідей Міжнародної науково-технічної конференції молодих учених та студентів, 19–20 грудня 2012 р., м. Тернопіль – Тернопіль: Вид-во ТНТУ ім. Івана Пулюя, 2012. – С. 187-188.
2. Чабанюк Я.М. Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проекту / Я.М. Чабанюк, В.С. Яковина, Д.В. Федасюк і ін. // Інженерія програмного забезпечення. – Вип. 1. – 2010. – С. 1-6.
3. Половко А.М. Основы теории надежности / А.М. Половко, С.В. Гуров – СПб.: БХВ-Петербург, 2006. – 704 с.
4. Яковин В. Огляд основних підходів до аналізу надійності програмного забезпечення / В. Яковина, В. Смірнов // Комп’ютерні науки та інформаційні технології – Вип. 719. – 2011. – С. 278-282.
5. Локазюк В.М. Надійність, контроль, діагностика і модернізації ПК: Посібник / В.М. Локазюк, Ю.Г. Савченко. – К.: Видавничий центр “Академія”, 2004. – 376 с.