

А. А. ЗЕМЛЯНОЙ, ассистент кафедры АСУ НТУ «ХПИ»

ЭКСПЕРИМЕНТАЛЬНОЕ ПРИМЕНЕНИЕ МОДЕЛЕЙ И ТЕХНОЛОГИЙ ОБЕСПЕЧЕНИЯ КАЧЕСТВА ДЛЯ РАЗРАБОТКИ КОМПОНЕНТНОГО ПРОГРАММНОГО РЕШЕНИЯ РЕАЛЬНОГО ВРЕМЕНИ

Розглядається технологія забезпечення деяких характеристик якості компонентного програмного рішення реального часу. Дається приклад експериментального використання запропонованих технологій, який включає в себе визначення архітектури та вимог, використання прототипу для збору даних, аналіз результатів.

Ключові слова: забезпечення якості, технологія прототипування, компонентне програмне рішення реального часу.

Рассматривается технология обеспечения некоторых характеристик качества компонентного программного решения реального времени. Дается пример экспериментального использования предложенных технологий, который включает в себя определение архитектуры и требований, использование прототипа для сбора данных, анализ результатов.

Ключевые слова: обеспечение качества, технология прототипирования, компонентное программное решение реального времени.

A technology for quality assurance of some real-time software quality attributes is considered. An experimental example of applying suggested technologies is provided. The example includes defining the solution architecture and quality requirements, using the prototype for statistics collection, analysis of the results.

Keywords: quality assurance, prototyping technology, real-time software solution.

Введение. В вопросах качества программного обеспечения (ПО) как характеристики степени его соответствия поставленным требованиям важную роль играют не только функциональные, но также и нефункциональные требования. Согласно SWEBOK (Software Engineering Body of Knowledge) [1] выделяются такие основные атрибуты качества ПО как: функциональность, эффективность, надежность, сопровождаемость, удобство использования. В зависимости от типа разрабатываемой системы важность приобретают те или иные атрибуты качества [2]. Для систем реального времени, помимо функциональности, важны скорость обновления информации, объемы передаваемых данных за единицу времени, полнота отображаемых данных и минимальные их потери. Эти требования относятся к характеристикам временной эффективности, эффективности использования ресурсов, устойчивости к отказам и восстанавливаемости [3].

Проблема обеспечения характеристик производительности и надежности. Для обеспечения характеристик качества в настоящее время существуют различные технологии обеспечения качества (ТОК) [4]. Проблема использования существующих ТОК состоит в том, что они

преимущественно предназначены для моделирования функциональности, либо же основаны на аналитических математических моделях, которые проблематично построить для сложных систем. Поэтому актуальным является вопрос использования прототипов для обеспечения заданных характеристик надежности и производительности при фиксированных ресурсах. Для компонентных программных систем реального времени (КПР РВ) были разработаны специальные модели и ТОК, которые позволяют задать требования нефункционального характера, определить эффективность применения той или иной ТОК в зависимости от сложности системы, провести моделирование с помощью выбранной технологии и проанализировать полученные результаты с точки зрения обеспечения качества. Основным элементом комплекса является технология прототипирования [4]. Рассмотрим применение данной технологии для построения конкретного КПР РВ.

Определение архитектуры и требований разрабатываемого решения. Предположим, что требуется разработать КПР РВ для процессов мониторинга и управления промышленным оборудованием при помощи микроконтроллеров различных типов (Atlas, Sirius, PDU), которые поддерживают различные протоколы (МК01, МК02, Modbus соответственно). Все контроллеры подключаются к центральному серверу данных через адаптеры TCP/IP. Сервер данных производит регулярный опрос устройств и предоставляет интерфейс доступа к данным реального времени для клиентских приложений. Клиентские приложения предполагаются трех типов: 1) клиент архивирования данных, который с заданной периодичностью запрашивает данные реального времени и архивирует их в базе данных, 2) клиент администрирования, который позволяет не только просматривать, но и изменять конфигурацию и параметры всех устройств в системе, 3) клиент оператора, который позволяет просматривать текущие данные и выполнять команды в удаленном режиме, в том числе и через сеть Интернет.

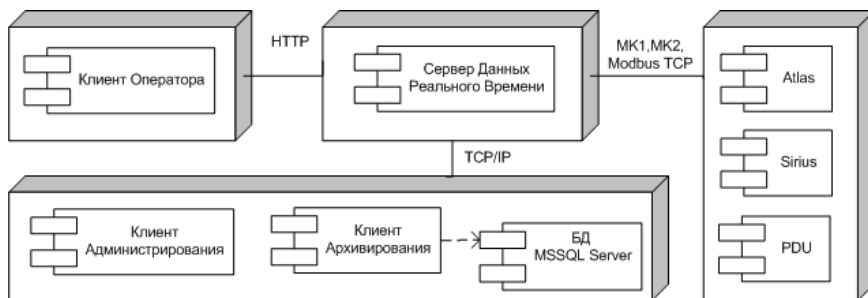


Рис. 1 – Диаграмма размещения компонентов исследуемого решения

На рис. 1 показана диаграмма размещения компонентов проектируемого решения. К основным характеристикам производительности и надежности

относится среднее значение реактивности $\bar{r} = \frac{1}{I} \sum_{i=1}^I (t_{2i} - t_{1i})$ (где t_{1i} – время отправки запроса, t_{2i} – время получения ответа, I – количество запросов); средняя производительность $\bar{P} = \frac{N}{T}$ (где N – количество ответов на запросы за время функционирования T); средняя наработка на отказ $\bar{T}_o = \frac{T}{N_o}$ (где T – время функционирования, N_o – количество отказов); интенсивность отказов $a = \frac{N_o}{N}$ (где N – число запросов, N_o – количество отказов). Интегральная

оценка соответствия требованиям вычисляется как $Q = \frac{1}{mQ_{\max}} \sum_{i=1}^m Q_i$ (где Q_i – оценка i -й характеристики, m – число оценок, Q_{\max} – максимальное значение оценки. Допустимый интервал для каждой характеристики определяется методом опроса нескольких экспертов и анализа их ответов.

После задания ограничений для значений характеристик качества необходимо оценить эффективность применения той или иной ТОК. Для этого была разработана методика оценивания [5], согласно которой экспертами оценивается сложность разрабатываемой системы и уровень требований качества, а затем, исходя из результатов, определяется эффективная для данного случая ТОК.

Прототипирование и анализ результатов. Процесс прототипирования проходит в несколько этапов [6]: реализация компонентов согласно построенной архитектуре, конфигурирование, выполнение и сбор статистики, обработка собранной статистики и анализ полученных результатов. Перед запуском прототипа выполняется конфигурирование компонентов под условия используемых протоколов и заданных требований. Во время выполнения созданного прототипа происходит накопление статистики при помощи компонента для сбора данных. Данные во время выполнения компонентов поступают в режиме реального времени и сохраняются в базе данных. Поскольку с реальными микроконтроллерами выполнять эксперименты зачастую не представляется возможным, то в процессе прототипирования они заменяются эмуляторами. Под эмулятором понимается ПО, которое в точности реализует логику работы и интерфейс микроконтроллера и в экспериментальных целях может быть использовано вместо физического микроконтроллера. При накоплении достаточного объема статистики выполнение прототипа останавливается и производится следующий шаг – анализ результатов. На основе собранных данных вычисляются значения искомым характеристик, путем сравнения с заданными ранее допустимыми интервалами определяются оценки

соответствия. Требование считается выполненным в полной мере и получает максимальную оценку соответствия тогда, когда измеренное значение характеристики принадлежит допустимому интервалу. По мере удаления от допустимого интервала уменьшается оценка соответствия. На рис. 2 представлены программные компоненты технологии прототипирования.

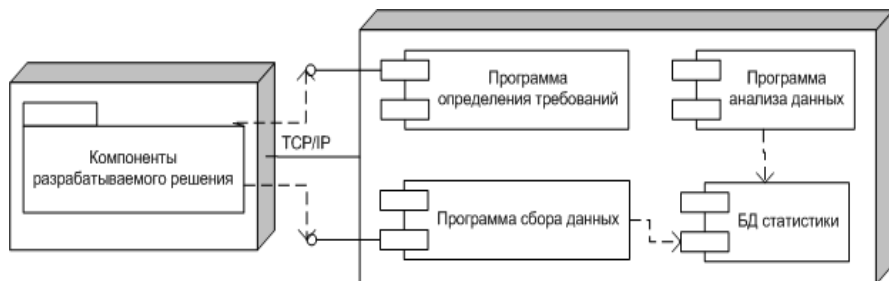


Рис. 2 – Программные компоненты технологии прототипирования

В рамках данного эксперимента программная реализация прототипа, как и конечного решения, осуществляется на платформе Microsoft .NET [7]. В частности, сервер данных реализован как Windows Service, клиент архивирования также, в качестве СУБД используется MS SQL Server 2012, клиент администрирования реализован как приложение Windows Forms, а клиент оператора как приложение MVC Web с использованием jQuery для асинхронного обновления данных. Исследуем характеристики реактивности и интенсивности отказов для каждого типа клиента (1 – клиент архивирования, 2 – клиент администрирования, 3 – клиент оператора).

Таблица 1 – Результаты расчета характеристик

Параметр (единица измерения)	Допустимый интервал	Измеренное значение	Оценка соответствия
Средняя реактивность 1 (с)	[0; 3]	0,536	9
Средняя реактивность 2 (с)	[0; 0,5]	0,501	7
Средняя реактивность 3 (с)	[0; 1]	1,945	1
Интенсивность отказов 1	[0; 0,1]	0,016	9
Интенсивность отказов 2	[0; 0,1]	0,020	9
Интенсивность отказов 3	[0; 0,2]	0,21	7

По результатам расчетов, показанных в табл. 1, одна из характеристик в исследуемой конфигурации значительно выходит за рамки допустимого интервала (средняя реактивность 3, клиент оператора) и имеет минимальную оценку соответствия, равную 1. Интегральная оценка соответствия имеет

значение $Q = 0,77$ при максимальном значении равном 1. Для повышения степени соответствия требованиям был проведен анализ программной реализации компонентов и найдена причина, которая лежала в низкой производительности сценариев JavaScript, используемых в клиенте для асинхронной загрузки данных. Для повышения производительности технология реализации клиента оператора ASP.NET MVC и jQuery была заменена на Silverlight, которая специально предназначена для построения сложных клиентских приложений, работающих в обозревателе. В результате повторного эксперимента с новым компонентом клиента оператора на платформе Silverlight было получено значение средней реактивности клиента равное 0,898. В такой конфигурации решение имеет интегральную оценку соответствия равную $Q = 0,93$ и в высокой степени удовлетворяет требованиям производительности и надежности, а прототип может быть использован как основа для разработки полнофункционального продукта.

Выводы. В ходе эксперимента были исследованы некоторые характеристики производительности и надежности проектируемого КПР РВ при помощи предложенной технологии прототипирования и сделаны выводы о соответствии оценок характеристик поставленным требованиям. Рассмотренная технология имеет практическую ценность, поскольку позволяет обеспечить требуемые характеристики качества ПО на ранних стадиях разработки, что в свою очередь снижает затраты на реализацию программного решения.

Список литературы: 1. Офіційний Інтернет-ресурс консорціуму SWEBOK: www.swebok.org. 2. Андон Ф. И. Основы инженерии качества программных систем / Ф. И. Андон, Г. И. Коваль, Т. М. Коротун и др. – 2-е изд. – К. : Акадамперіодика, 2007. – 580 с. 3. Липаев В. В. Методы обеспечения качества крупномасштабных программных средств. – М. : СИНТЕГ, 2003. – 520 с. 4. Земляной А. А. Комплекс моделей технологии прототипирования для адаптивного проектирования компонентных программных решений / А. А. Земляной, Н. В. Ткачук, Р. А. Гамзаев // Вісник Національного технічного університету «ХПІ». – Харків : НТУ «ХПІ». – 2008. – № 5. – С. 97–107. 5. Земляний А. О. Моделі та засоби експертної оцінки ефективності технологій забезпечення якості програмних систем реального часу / А. О. Земляний, М. В. Ткачук // Радіоелектронні і комп'ютерні системи. – Харків : Національний аерокосмічний університет ім. М.Є. Жуковського «Харківський авіаційний інститут». – 2009. – № 1. – С. 67–72. 6. Земляной А. А. Комплексная методика прототипирования компонентных программных систем на основе технологий обеспечения качества / А. А. Земляной, Н. В. Ткачук // Вестник НТУ «ХПИ». – Харьков : НТУ «ХПИ». – 2010. – № 67. – С. 110–115. 7. Рихтер Д. Программирование на платформе Microsoft .NET Framework. – М. : Русская Редакция, 2002. – 486 с.

Надійшла до редколегії 11.12.2012