ПРОГРАММА ДЛЯ ФОРМИРОВАНИЯ СОСТАВА МУЛЬТИВЕРСИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Е.В.Соловьев

Институт космических и информационных технологий СФУ

В настоящее время существуют два взаимно дополняющих друг друга подхода для разработки устойчивого к сбоям ПО: безопасное программирование и отказоустойчивые системные архитектуры. Нас интересует подход с использованием отказоустойчивых архитектур, в его основе лежит резервирование аппаратных и программных средств и выявление ошибочных результатов, посредством сравнения данных.

В большинстве случаев технология разработки устойчивого к сбоям оборудования, базируется на методе модульного резервирования, когда модуль оборудования дублируется несколько раз. Выходные данные каждого модуля сравниваются между собой: если один из модулей выходит из строя или его выходе данные отличаются от выходных данных других модулей, эти данные игнорируются. Если сразу восстановить сбойный модуль невозможно, система автоматически отключает его, и продолжает свою работу с оставшимися модулями.

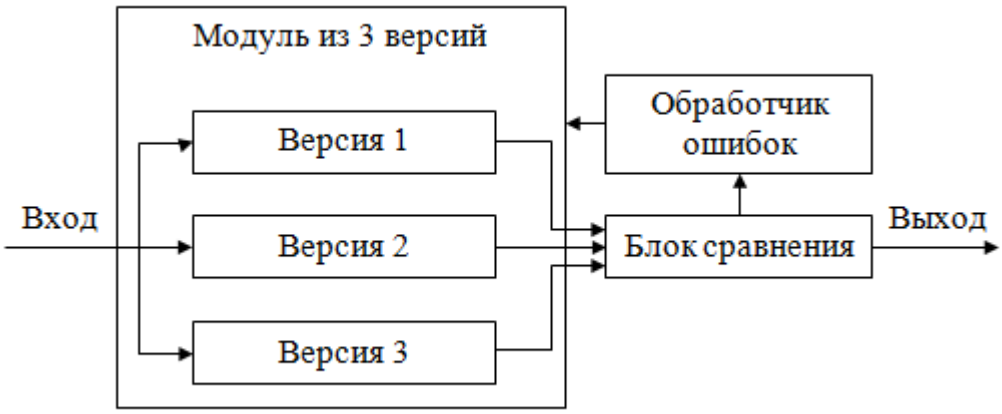

Рисунок 1. Мультиверсионный модуль

Существует две методологии для создания отказоустойчивого программного обеспечения: блоки восстановления и мультиверсионное програмирование (*N*-version programming или multiversion programming). При использовании методологии мультиверсионного программирования осуществляется разработка мультиверсионного программного обеспечения (NVC - n-version software), которое строится на множестве версий с общей спецификацией, созданными разными командами разработчиков. Эти версии выполняются параллельно. Результат их работы сравниваются с помощью системы согласования, результаты версии, не совпадающие с

половиной версий и еще одной или не полученные вовремя, отвергаются. Это наиболее часто используемый подход к созданию отказоустойчивого программного обеспечении (Рисунок 1).

Отказоустойчивое программное обеспечение должно использоваться только в системах, критических по обеспечению безопасности, так как мультиверсионная методология усложняет создание и проверку программного обеспечения, а также значительно увеличивает его стоимость.

Так как улучшение характеристик надежности ПО с использованием избыточности требует дополнительных затрат, основной задачей, встающей перед разработчиком является достижение баланса между двумя противоречащими задачами, максимизировать надежность программного обеспечения и при этом сохранить стоимость системы в определенных границах.

Использована следующая модель для описания мультиверсионного ПО:

n – число модулей NVS - системы программного обеспечения;

m_i – число версий, i-го модуля, i = 1, ...,n;

R_{ij} – оценка надежности j-ой версии i-го модуля;

X_{ij}– булева переменная, равная 1, если j-я версия выбрана для i-го модуля, иначе - 0;

g_{ij} – вероятность перехода из i-го модуля в j-ый модуль;

R_i – оценка надежности i-го модуля;

R – оценка надежности NVS - системы ПО;

C_{ij} – стоимость разработки j-ой версии i-го модуля;

B - ограничение по стоимости создаваемой NVS - системы ПО.

Модель представляет из себя наиболее общий случай мультиверсионного программного обеспечения. Сформулируем задачу.

$$R \rightarrow \max R,\ C \rightarrow \min C, \tag{1}$$

$$R_{parallel} = \sum_{j \in S_i} g_{ij} R_j, \tag{2}$$

$$R_{\text{coherently}} = \prod_{i \in S_k} R_i\ , \tag{3}$$

при ограничениях

$$\sum_{j=1}^{m_i} X_{ij} \geq 1, i = \{1, \dots, n\}, \quad (4)$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} X_{ij} C_{ij} \leq B, \quad (5)$$

$$X_{ij} = \{0,1\}, i = \{1, \dots, n\}, j = \{1, \dots, m_i\}, \quad (6)$$

где R_i задаётся выражением

$$R_i = 1 - \prod_{j=1}^{m_i} \left(1 - R_{ij}\right)^{X_{ij}}. \quad (7)$$

На основе модели представления мультиверсионного ПО и алгоритма для его формирования, был реализован программный комплекс, получившем название, ACO N-version software creator. Данное программное приложение было написано на языке программирования C# с использованием среды программирования приложений Microsoft Visual Studio 2012. Это позволило использовать возможности, предоставляемые объектно-ориентированным подходом при разработке программного обеспечения, а также задействовать Windows Forms Framework, при разработке графического интерфейса пользователя, знакомого любому пользователю персонального компьютера, работающего под управлением операционной системы Windows.

В итоге была создана программа, схема которой представлена на рисунке 2. Она позволяла проводить эксперименты на задачах формирования оптимального состава мультиверсионного программного обеспечения.

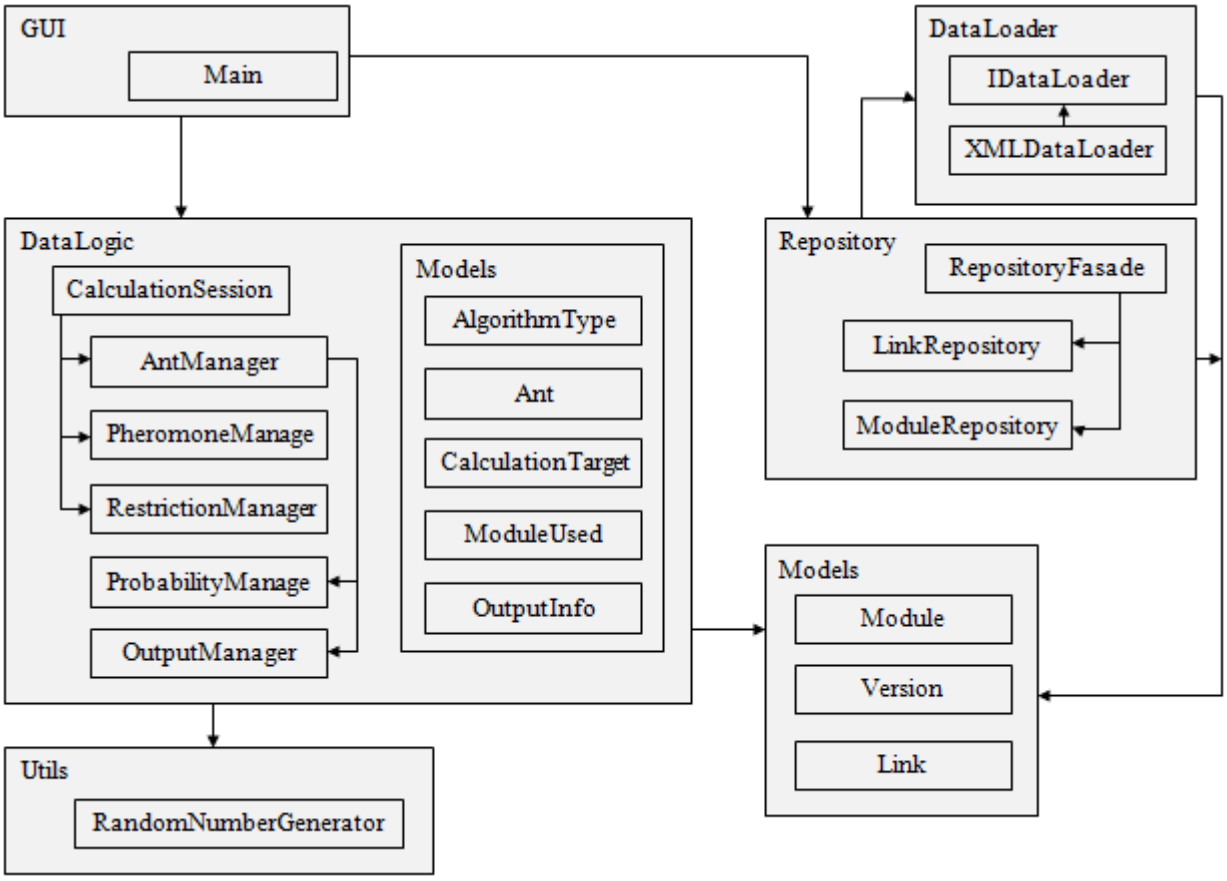

Рисунок 2. Схема программы

В результате создания программной реализации алгоритма, был получен инструмент, с помощью которого был осуществлен ряд экспериментов, для получения данных, с помощью которых было произведено расчет состава мултиверсионного ПО на тестовой задаче. В результате анализа полученных результатов было установлено, что программа способна рассчитывать оптимальный состав программного обеспечения, с учетом задаваемых ограничений.

Список литературы:

1. Ковалёв, И. В. Мультиверсионный метод обеспечения отказоустойчивости программных архитектур систем управления критичными по надёжности техническими объектами / И.В. Ковалёв, Р.В. Юнусов // Науч. конф. «Научно-инновацинное отрудничество». Москва, 2002. – 172 с.
2. Berman, O. Optimization models for reliability of modular software systems / O.Berman, N.Ashrafi // Software Engineering, 1993. p. 1119 - 1123.