УДК 519.8

ОГРАНИЧЕНИЯ НА СОВМЕСТИМОСТЬ ВЕРСИЙ В ЗАДАЧЕ ФОРМИРОВАНИЯ СТРУКТУРЫ МУЛЬТИВЕРСИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ерыгин В. Ю.,
научный руководитель д-р техн. наук Ковалев И. В.
Сибирский государственный аэрокосмический университет имени академика М.Ф. Решетнева

На сегодняшний день высокие темпы развития технического прогресса позволяют информационным технологиям широко охватывать практически все сферы деятельности человека, что ведёт к её несомненному улучшению, однако, решая одни проблемы, мы сталкиваемся с другими. Создание систем, охватывающих различные сферы деятельности человека весьма трудоёмкий процесс, вызванный проникновением в нашу жизнь различных технологий. Это неизменно, приводит не только к повышению сложности, но и снижению надежности подобных систем. В связи с этим остро встает вопрос не просто о создании различных информационных систем, систем управления, или систем поддержки принятия решений, а в первую очередь систем повышенной надежности.

Общая надежность системы складывается из надежности программной и аппаратных частей. При этом повлиять на надежность аппаратной части довольно затруднительно в то время, как надежность программной составляющей всецело зависит от разработчиков. Таким образом перед проектировщиками информационных и управляющих систем встает задача включения в неё программной части, которая обладает повышенной надежностью.

Для решения этой задачи всё большее распространение среди разработчиков получает зарекомендовавшая себя методология мультиверсионного программирования. Предложенная еще в 1976 году А. Авиженисом методология мультиверсионного проектирования программных средств позволяет значительно повысить их надежность за счет введения программной избыточности. Данный подход нашел широкое применение при построении программных систем с требованием высокого уровня надежности, т.е. отказоустойчивых систем, которые используются в таких сферах деятельности человека, как космос, авиация или атомные энергостанции и т.п.

Однако при формировании мультиверсионного программного средства возникает ряд проблем, с которыми приходится сталкиваться разработчику. Отталкиваясь от основного требования создать надежную и отказоустойчивую систему, приходится учитывать одно из главных ограничений - ограничение на выделяемые средства. С возникновением подобных трудностей задача постоения мультиверсионного программного средства сводится к решению задачи математического программирования.

Для решения этой задачи могут применяться хорошо зарекомендовавшие себя методы многоатрибутивного принятия решений, однако, не смотря на то, что они хорошо себя зарекомендовали в решении подобных задач, зачастую в реальности возникают ограничения, которые невозможно учесть в модели, используемой этими методами.

Часто на практике возникает ситуация, когда невозможно заранее точно предсказать требуемый объем финансирования на разработку той или иной системы. Тогда речь идет о нечетком бюджете. Помимо увеличение сложности программных систем, а тем более применение мультиверсионного подхода, влечет за собой усложнение связей между различными частями программы, а следовательно и невозможность в один и тот же момент времени использовать определенный набор компонентов программного средства. В этом случае говорят об ограничениях на совместимость версий. Появление этих ограничений обусловлено самыми различными факторами, учет которых приводит к усложнению структуры ограничений. Представив ограничения в виде деревьев, можно выделить несколько классов в зависимости от глубины их связей. При этом на каждом уровне глубины дерева может быть произвольное количество элементов. Таким образом класс ограничений варьируется от единицы и до числа модулей в программном средстве N, а количество элементов на каждом уровне варьируется в пределах от одного и до $m_i - 1$, $i = \overline{1, N}$ где m_i – число версий, доступных для каждого модуля.

Подобного рода ограничения возникают на этапе формирования архитектуры программных средств. Учесть их в модели, а так же получить решение, используя существующие подходы на сегодняшний день невозможно. Помимо этого, увеличение сложности программного обеспечения ведёт к необходимости изменения структуры в режиме реального времени. Эти изменения касаются, как самой структуры программного средства (число модулей и версий доступных каждому модулю), так и параметров, входящих в неё компонентов. Учет такого поведения требует введения понятия динамической мультиверсионности. Поскольку ограничения на совместимость версий при формировании структуры мультиверсионного программного обеспечения возникают вне зависимости от этапа на котором происходит формирование (или изменение) структуры, то учет их в режиме реального времени позволит существенно расширить область применения методологии мультиверсионного программирования, а так же усложнить программную составляющую управлющих систем без ущерба надежности. Таким образом, с применением предлагаемого подхода, появляется возможность учета различных типов ограничений на совместимость версий модулей не только на этапе формирования структуры, но и в режиме реального времени, когда отсутствует возможность решения задачи методом полного перебора.

Опираясь на, предложенный в 1970 году подход Беллмана Заде, а так же принцип оптимальности динамического программирования Беллмана, предложен алгоритм построения структуры мультиверсионного программного обеспечения с учетом вышеописанных ограничений. Представляя множества ограничений и целей, как нечеткие подмножества множества альтернатив, и разбивая задачу выбора на этапы, согласно принципу динамического программирования Беллмана, мы находим решение на пересечении нечетких множеств целей и ограничений. При этом данный подход позволяет учитывать ограничения на совместимость.

Таким образом предложенный подход формирования структуры мультиверсионного программного обеспечения позволяет решить такие основные проблемы, как проблема нечеткости бюджета и совместимость версий. Наряду с этим обеспечивается возможность формирования структуры программных средств в режиме реального времени, что в свою очередь расширяет возможную область применения методологии мультиверсионного программирования.