

ПРИМЕНЕНИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ УЛУЧШЕНИЯ КАЧЕСТВА ПРОГРАММ

В.В. Бураков (Санкт-Петербург)

В настоящее время в разработку программ вовлечено большое число людей и организаций, на их создание расходуются огромные ресурсы. Программы проверяются на соответствие в основном функциональным требованиям, адекватная оценка выполнений требований к качеству, равно как и полнота их описания, вследствие отсутствия общепринятых спецификаций качества программ и автоматизации их проверок, затруднены [1]. Это неизбежно отражается на снижении качества программных продуктов, приводит к потерям разного характера, порождает опасность для жизни людей и планеты. Важность задач, решаемых программным обеспечением и упомянутые пробелы в современном состоянии программной инженерии, обосновывают высокую актуальность исследований, направленных на разработку математических моделей, методов и алгоритмов для формализации процессов оценивания и управления качеством программ. Учитывая сложность семантики кода и трудно предсказуемую динамику его изменения в процессе жизненного цикла, для управления качеством программных средств целесообразно использовать имитационное моделирование.

Модель программ

Для моделирования программы в контексте улучшения ее качества использовались элементы теории графов, дополненные [2-6]:

- метками дуг и вершин, для идентификации элементов программ и связей между ними;
- ролями вершин для описания способов использования программных сущностей - классов, методов, атрибутов, переменных и т.п.;
- типами для отображения классов элементов программы;
- номерами дуг для задания порядка передачи управления от одной программной структуры к другой;
- вето-графами, которые представляют собой абстрактные графы, дополненные регулярными выражениями, и используются для задания ограничений на возможные связи между элементами программы.

Модель программных средств была разработана обобщенной. Для ее использования необходимо осуществить настройку: выделить все виды программных элементов и отношений между ними, которые влияют на оценку качества, определить соответствующие типы и роли вершин и дуг в графовой модели, задать вето-графы для определения ограничений.

Существенным плюсом, облегчающим использование модели, является естественность и простота ее визуального представления. На рисунке 1 приведен пример визуализации объектно-ориентированного шаблона «Наблюдатель». Типы вершин - в первой секции прямоугольника, метки – во второй, в третьей – роли, у дуг – только типы и метки.

Важной особенностью предлагаемой техники моделирования свойств программ является возможность варьировать составы типов программных сущностей и связей между ними, подбирать их в соответствии с задачами оценивания и управления качеством, добиваясь наилучшего компромисса между адекватностью модели и ее эффективностью.

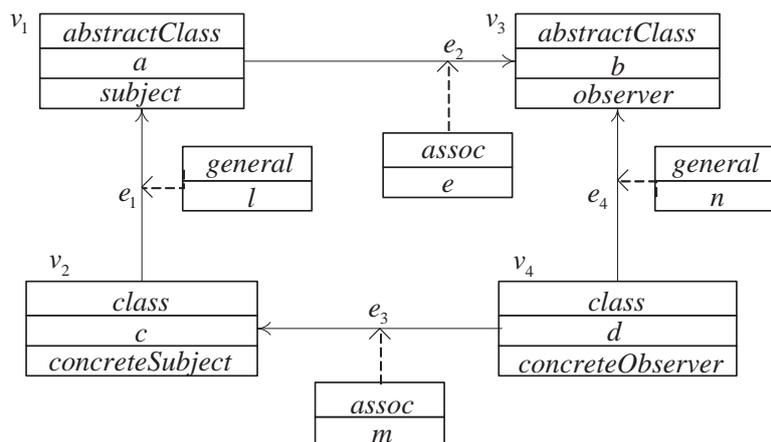


Рис.1. Пример представления шаблона «Наблюдатель»

Моделирование качества программ

Для создания модели качества программ была осуществлена систематизация и обобщение основных подходов к описанию качества программных средств. Основной вектор развития моделей качества определяется попытками сделать все атрибуты качества измеримыми, а также стремлением разнести на разные уровни качественные и количественные атрибуты. Основными недостатками существующих моделей являются:

- неформализованный характер - практически все модели математически не проработаны, поэтому отсутствует возможность доказательств корректности моделей или формализованного описания каких-либо их свойств;
- терминологическая несогласованность - параллельно существует и используется несколько интерпретаций показателей качества и метрик в разных моделях качества;
- трудность технологической интерпретации – нет четких зависимостей между показателями качества и практикой разработки программных продуктов.

При разработке модели качества основное внимание уделялось математически строгому описанию качественных свойств, независимости от типов программ, согласованности с основными существующими моделями и стандартами, обеспечению связи с конструктивными свойствами. В качестве основы модели использовалась теория категорий, обладающая мощными выразительными средствами и позволяющая создавать универсальные конструкции.

Класс объектов категории качества (q) представляет концептуальные понятия, характеризующие качество программ (например, показатели качества, принципы проектирования и т.п.). Класс морфизмов категории качества описывается тремя множествами: множеством одноуровневых ($\bar{\xi}$), строгих (ξ) и нестрогих ($\tilde{\xi}$) иерархических морфизмов. Для иерархических морфизмов принципиальное значение имеет направленность - от объектов, находящихся на верхних уровнях иерархии, к объектам, расположенным на нижних. Эта направленность определяет главное назначение модели качества - последовательное, детализируемое на каждом следующем уровне иерархии, концептуальное описание понятия качества программы.

Модель качества (рис. .) представляет собой подкатегорию категории качества, состоит из конечного числа объектов категории качества и конечного числа морфизмов между ними. Модель может представлять некоторый международный или государственный стандарт, стандарт предприятия-разработчика и т. п. В подкатегорию категории качества выбираются объекты, представляющие концепции качества, исходя из назначения модели.

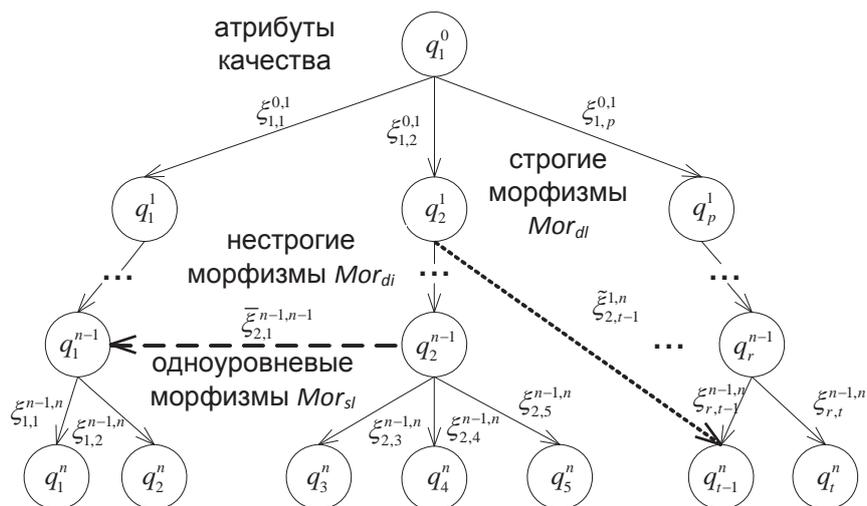


Рис. 2. Схема модели качества программ

Модель качества способна математически строго выразить необходимые свойства, предъявляемые к качеству и обеспечить их формальную верификацию. Так, с помощью модели можно обеспечить строгую древовидную структуру для гарантии отсутствия нежелательных зависимостей, выделить базисные слои для их повторного использования, обеспечить согласованность зависимостей показателей качества.

При формировании множества объектов категории качества, необходимо иметь такой состав атрибутов качества, который бы покрывал все существующие и будущие запросы, поэтому это множество открыто.

Выбор объектов и морфизмов для включения в состав модели качества зависит от целей моделирования. Выбранные объекты приобретают контекстную зависимость и интерпретируются в соответствии с предметной областью, основными элементами которой являются парадигма программирования (объектно-ориентированная, функциональная, структурная и т.п.), язык программирования (java, c, lisp и т.п.), тип проекта (веб-приложение, клиент-серверное приложение с насыщенным клиентом, приложение для смартфона и т.п.). Процессы формирования множества объектов и морфизмов для категории качества, выбор объектов и морфизмов для синтеза модели качества программы - работа экспертов. При этом последним, помимо профессиональной интуиции, при выполнении этих видов работ необходимо использовать математический аппарат, заложенный в основу модели качества программ.

Оценивание качества программ

Для математического описания процесса оценивания качества программ была разработана модель измерений. Основными требованиями при ее разработке явились: строгое математическое описание количественных свойств программы, независимость от типов программ, языков и технологий разработки, возможность анализа функциональных зависимостей метрик, обеспечение связей с моделями качества и преобразований. Математической основой модели измерений выступили теория категорий и элементы функционального анализа.

Категория измерений (рис. **Ошибка! Источник ссылки не найден.**) состоит из объектов - метрических пространств (M) и морфизмов - операторов комплексирования (A). Графовые модели программ образуют множества метрических пространств, метрики определяют обсчет элементов графовой модели. Для определения метрических пространств было сформулировано понятие слабой функциональной эквивалентности. Два программных средства считаются слабо функционально эквивалентными, если на один и тот же набор функциональных тестов оба средства выдают одинаковый результат. Каждое метрическое пространство содержит все математические модели программы, которые состоят в отношении слабой функциональной эквивалентности с другими моделями этого пространства. Между собой эти модели отличаются внутренней структурой, при этом, вещественное число - расстояние между программой x и программой y является метрикой этого пространства. Морфизмы категории измерений -

Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

операторы комплексирования, задаются на метрических пространствах и отражают зависимости одних производных метрик от других, производных или базовых. В категории измерений, в отличие от категории качества, морфизмы имеют противоположное направление. Это соответствует основному назначению этой категории - определять метрические пространства для оценки атрибутов качества и задавать функциональные зависимости для порождения значений производных метрик верхних уровней за счет комплексирования метрик нижних уровней иерархии.

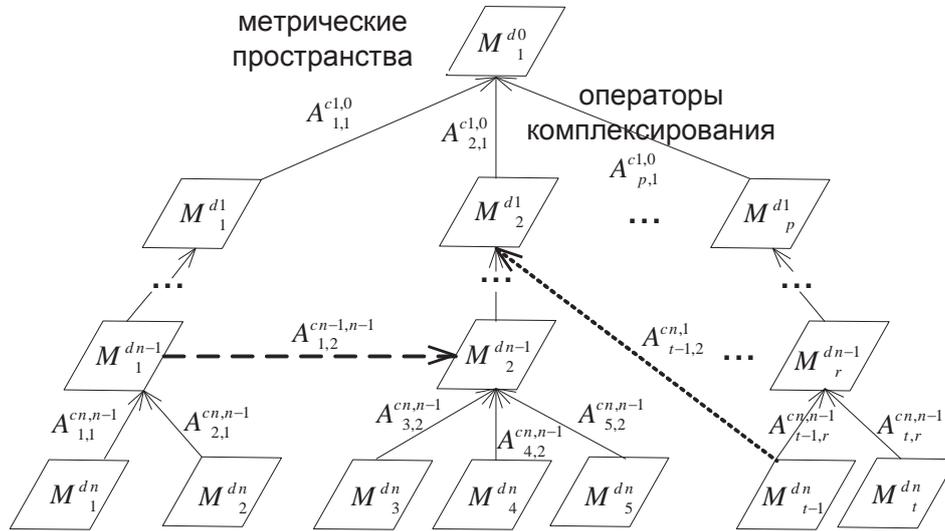
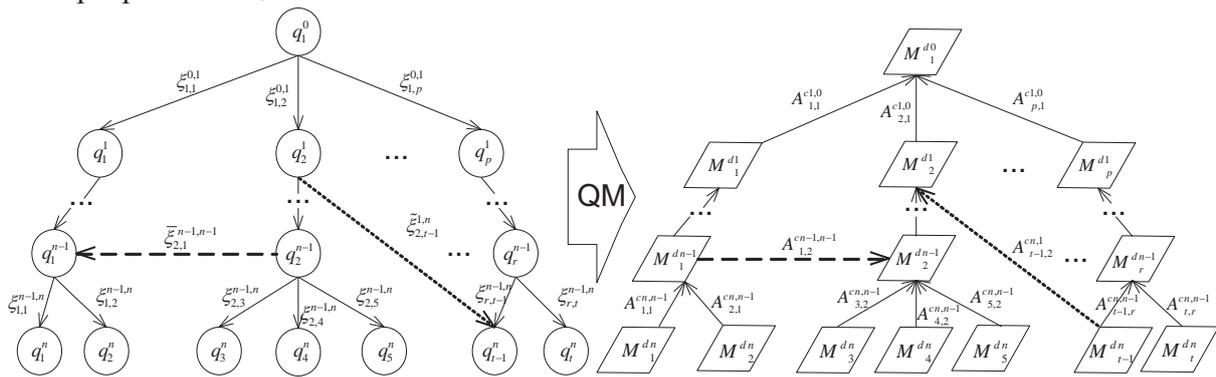


Рис.3. Схема модели измерений программ

Для генерации структуры категории измерений на основе категории качества был введен контравариантный одноместный функтор (QM) (рис. **Ошибка! Источник ссылки не найден.**). Он отображает объекты категории качества на метрические пространства категории измерений, а морфизмы категории качества - на морфизмы категории измерений. С помощью функтора можно количественно оценить элементы модели качества, как для конкретного показателя, так и для качества программы в целом.



$$QM\left(\bigcup_{i \in \{0, n-1\}} \{q^i\}\right) = \bigcup_{i \in \{0, n-1\}} \{M^{di}\} \quad QM(\xi_{s,m}^{i-1,i}) = A_{m,s}^{ci,i-1} \quad QM(\tilde{\xi}_{s,m}^{i,j}) = A_{m,s}^{cj,i} \quad QM(\bar{\xi}_{s,m}^{i,i}) = A_{m,s}^{ci,i}$$

Рис.4. Функтор для построения модели измерений

Для целенаправленного и согласованного использования моделей качества, измерений и программ в процессе оценивания качества, был определен метод оценивания качества. Его основными этапами являются:

- создание структуры модели измерений, повторяющей структуру модели качества с точностью до обращения морфизмов;
- подбор базовых и производных метрик для измерения листовых элементов;

Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

- определение типов вершин и дуг, которые будут достаточны для формализованного описания метрик, построение модели программного средства;
- формализованное описание метрик с помощью модели программ;
- задание операторов комплексирования, связывающих метрики разных уровней иерархии. Улучшение качества программ

Для улучшения качества программ была разработана модель преобразований. Основными требованиями к ней выступили необходимость строгого математического описания программных преобразований, независимость от типов программ и языков программирования, обеспечение связей с моделью программ и моделью их измерений. В качестве математической основы модели были выбраны условные графовые продукции (p), базирующиеся на теории графов и категорий. Описание продукции состоит в задании в левой части продукции тех вершин и дуг, которые должны иметься в исходном графе, а в правой - соответствующих им элементам графа после ее применения. Частичный морфизм описывает отношения объектов левой части к правой, детализируя процесс применения продукции. Благодаря использованию условий преобразований появляется возможность описать, в каких именно случаях исходные графы могут быть приведены к целевым. Базис преобразований следует из структуры графа, моделирующего программу, и состоит из операций добавления и удаления вершин и дуг каждого типа. Производные преобразования создаются путем суперпозиции базовых. Оба типа преобразований являются операторами, имеющими в качестве областей определений и значений подпространства метрических пространств категории измерений (рис. 5).

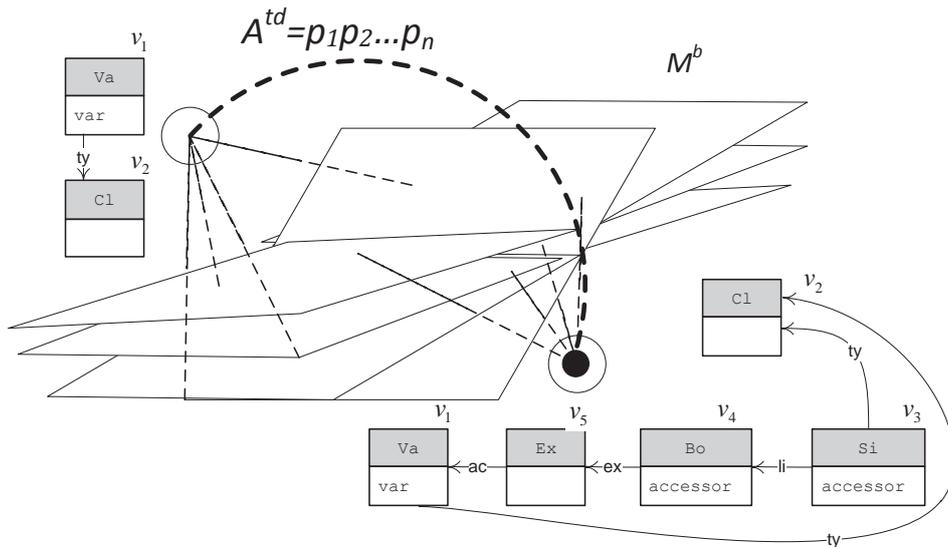


Рис.5. Схема улучшения качества программы

Любой из операторов может характеризоваться побочным эффектом - помимо изменения значения целевой метрики может изменить значения других метрик, ухудшив качество программы. Под оптимальным изменением программы понимается максимальная близость точки в многомерном метрическом пространстве, характеризующей текущее состояние программы к точке, соответствующей эталонному состоянию. Теоретически любая точка модели измерений достижима, проблема реальной достижимости возникает благодаря ограничениям на области определений и значений операторов метрических пространств, которые являются следствием необходимости учета семантики языка программирования и функциональных свойств программы.

В общем случае, задача выбора набора преобразований для улучшения качества, является многокритериальной под оптимальной задачей поиска точки в окрестности эталона. Учитывая сложность решения этой задачи в условиях существования побочных эффектов, а также ограничений, накладываемых языком программирования и функциональными требованиями, для нахождения оптимальной последовательности преобразований используются принципы

Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

имитационного моделирования. При таком подходе все наборы преобразований применяются над моделью программного средства. Каждая из итоговых моделей программных средств затем измеряется с помощью модели измерений, выбирается та из них, которая наиболее соответствует модели качества.

Структура имитационной модели

На рис. 6 приведена структура имитационной модели, задачами составляющих ее подсистем явились:

- подсистема моделирования кода: разработка структуры модели программы, определение типов и ролей, генерация модели программы на основе кода, генерация кода на основе модели;
- подсистема моделирования качества: моделирование качества, генерация модели измерений, определение состава базовых метрик, задание функциональных зависимостей между метриками, задание эталонных значений метрик, анализ репозитория проектов для автоматического определения эталонных значений метрик;
- подсистема моделирования измерений: описания качественных состояний, определение значений метрик на основе модели измерений и модели программы, определение состояния программы и ее компонент на основе значений метрик;
- подсистема моделирования преобразований: генерация базовых и задание производных преобразований, определение оптимального набора преобразований на основе имитационного моделирования их последствий.

Цифрами на рис. 6 обозначены основные этапы процесса улучшения качества исходного кода программ:

1: модель качества устанавливает критерии качества, значимые для конкретного проекта; на ее основе создается модель измерений;

2: подсистема моделирования измерений на основе модели качества и модели текущего состояния кода производит оценку качества;

3: данные о несоответствии исходного кода требуемому качеству поступают в подсистему моделирования преобразований, которая осуществляет подбор преобразований, способных улучшить качество кода;

4: происходит построение модели исходного кода с учетом найденной последовательности его преобразований;

2: новая модель кода поступает в подсистему моделирования измерений с целью учета побочных эффектов, связанных с его преобразованием.

Секция 3. Практическое применение моделирования и инструментальных средств автоматизации моделирования, принятие решений по результатам моделирования

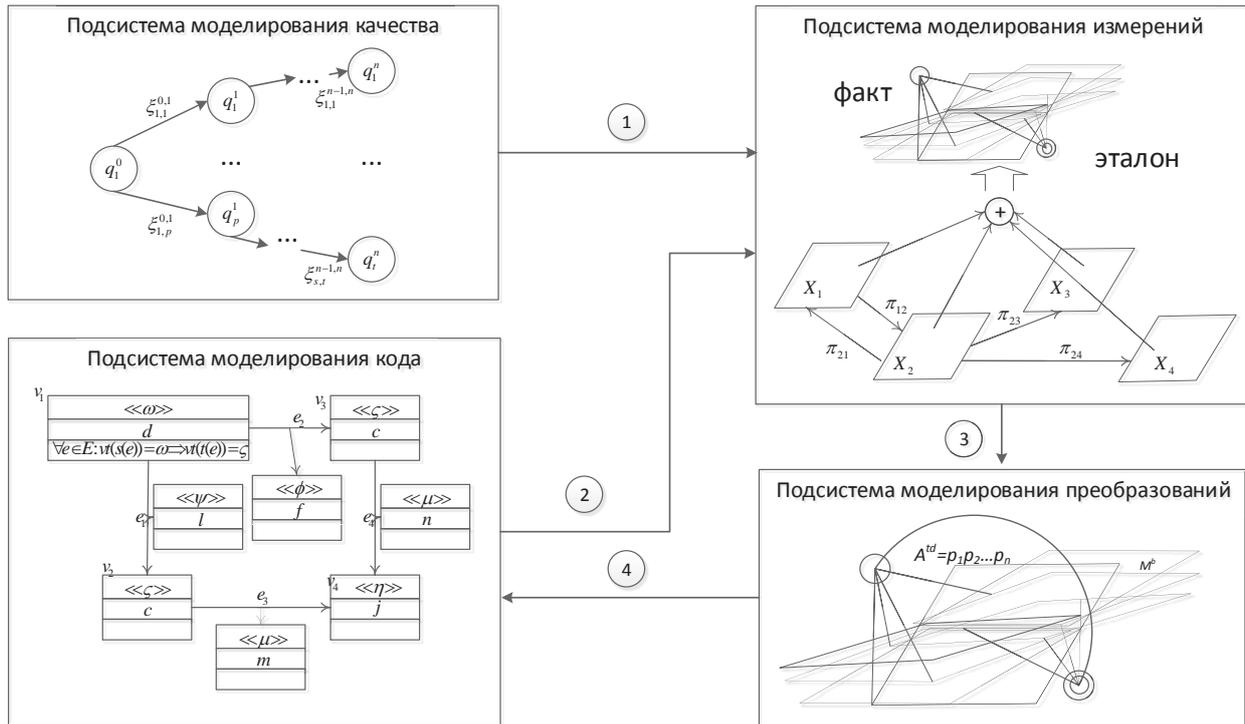


Рис.6. Схема имитационной модели улучшения качества

В рамках предлагаемого имитационного подхода к улучшению качества кода процессы оценивания качества, формирования преобразований, построения новой модели кода сменяют друг друга до тех пор, пока качество кода не будет отвечать искомому критерию. Найденная таким образом последовательность преобразований применяется затем к исходному коду программного средства.

Заключение

Применение представленного подхода, с учетом приведенных особенностей, позволит заложить основу для создания высококачественных программ и обеспечить экономию материальных и человеческих ресурсов, необходимых для их разработки.

Исследования, выполненные по данной тематике, проводились при финансовой поддержке гранта РФФИ № 17-08-00797 и госбюджетной темы № 0073–2019–0004.

Литература

1. Липаев В.В. Выбор и оценивание характеристик качества программных средств, М.: Синтез, 2001 г., 228 стр.
2. Бураков В.В. Управление качеством программных средств. Монография. Санкт-Петербургский государственный университет аэрокосмического приборостроения, 2009, 287 стр.
3. Бураков В.В. Управление качеством программных средств. // Информационно-управляющие системы № 5, 2009, стр. 43-47.
4. Бураков В.В. Оценка качества программных средств. // Авиакосмическое приборостроение № 4, 2009, стр. 28–33.
5. Бураков В.В. Система управления качеством программных средств. // Авиакосмическое приборостроение № 5, 2009, стр. 27 – 31.
6. Бураков В.В. Концептуальное моделирование качества программных средств // Авиакосмическое приборостроение № 7, 2008, стр. 54–60.