

УДК [004.05 + 006.015.5 + 004.414.38] : 519.171.1

МЕТОД ПОВЫШЕНИЯ КАЧЕСТВА СПЕЦИФИКАЦИИ ТРЕБОВАНИЙ К ПРОГРАММНОМУ СРЕДСТВУ

А.Ю. ЧИРКОВА, В.В. БАХТИЗИН

*Белорусский государственный университет информатики и радиоэлектроники
П. Бровка, 6, Минск, 220013, Беларусь*

Поступила в редакцию 27 марта 2015

Предложена графовая модель спецификаций требований. При построении графовой модели описаны действия, которые позволяют обеспечить соответствие требований критериям качества, что в совокупности представляет собой метод повышения качества спецификации требований к программному средству.

Ключевые слова: качество спецификаций требований, требования к программному средству, управление требованиями, граф.

Введение

Тщательно разработанные требования к программному средству (ПС) являются ключевым моментом, влияющим на успех проекта. Стоимость исправления ошибки после завершения проекта на несколько порядков выше, чем расходы по исправлению подобной ошибки на этапе анализа требований. Требования – это описанные возможности, которыми должно обладать ПС, а также ограничения, накладываемые на ПС [1]. Очевидно, чем сложнее разрабатываемое ПС, тем сложнее структурировать и управлять требованиями.

Управление требованиями представляет собой деятельность, где параллельно происходит сбор требований, их анализ, документирование и проверка. Это также процесс мониторинга и документирования изменений в требованиях, отслеживание их изменений в сопутствующих документах или программных продуктах, и доведения этой информации до проектной команды [1].

Хорошо налаженное управление требованиями гарантирует, что непредвиденными изменениями можно управлять в течение всего жизненного цикла программного продукта. Иначе высокого качества программного продукта невозможно достичь. Однако, несмотря на важность обеспечения качества спецификаций требований (СТ), этой проблеме уделяется мало внимания в настоящий момент. Существуют определенные техники выявления требований [2], методы моделирования требований, например, семантические методы [2], объектно-ориентированные методы [3], а также рекомендации и стандарты к содержанию СТ. Однако модели и методы, позволяющие оценивать и повышать качество требований, в литературе описаны недостаточно полно.

Критерии качества требований

Критерии качества требований по-разному определены различными источниками. Анализируя ряд стандартов, таких как ISO/IEC 9126-1:2001 [4], IEEE 830-1998 [5], ISO/IEC 25010:2011 [6], можно выделить общие критерии качества программных продуктов и требований к ним. Наиболее весомыми из них являются следующие критерии: однозначность, полнота, непротиворечивость, атомарность, трассируемость, актуальность, выполнимость,

недвусмысленность, обязательность (важность), проверяемость, изменяемость (стабильность), целостность, корректность.

Конкретное требование может соответствовать тому или иному критерию качества, при этом, качественное требование должно отвечать всем перечисленным выше критериям качества. Спецификация требований высокого качества должна состоять только из качественных требований.

Однако если такие критерии как выполнимость или недвусмысленность легко проверить, то определить соответствие требования такому критерию как полнота довольно сложно. Поэтому этот критерий можно разбить на два типа.

1. Соответствие первоначально заявленным требованиям, и/или описанным бизнес-требованиям, а также наличие первоначально не заявленных, но необходимых для реализации или функционирования ПС, требований

2. Описание всех аспектов, включающих функциональные возможности, условия выполнения, ограничения проектирования, характеристики, внешние интерфейсы, реакции ПС на как допустимые, так и недопустимые значения входных данных во всех возможных ситуациях.

Построение графовой модели спецификации требований

Первоначальный набор требований не является формализованной структурой, но представляет собой механизм взаимодействия сторон, т.е. заинтересованных лиц, вовлеченных в процесс разработки ПС. В связи с тем, что количество заинтересованных лиц достаточно велико, кроме того, они отвечают за разные этапы разработки ПС, выделяют различные уровни требований.

Бизнес-требования содержат высокоуровневые цели заказчиков ПС. Требования пользователей описывают цели и задачи, которые пользователям позволит решить разрабатываемое ПС. Системные требования описывают требования к системе, которая в общем случае состоит из аппаратных и программных средств, а также ручных операций. Исходя из этого, первоначальный набор требований может быть представлен в виде множества предписаний или ожидаемых возможностей ПС: $R_{init} = \{r_i = \overline{1, n}\}$, где n – количество предписаний, r_i – i -ое предписание.

Сюръективное отображение $f: R_{init} \rightarrow R$ при котором каждый элемент множества R является образом хотя бы одного элемента множества R_{init} , представляет собой функцию, которая преобразовывает первоначальный набор предписаний в формализованные требования. Сюръекция обеспечивает полноту первого типа, так как ни одно первоначальное предписание не должно быть опущено, наоборот, сходные предписания могут быть формализованы в одно требование.

Преобразованное множество R требований можно описать в иерархическом виде так, чтобы они были сгруппированы по общему тематическому признаку или по назначению. Будем использовать понятие «домен» для определения таких групп. Таким образом, СТ может быть представлена в виде древовидной структуры, где листьями дерева являются непосредственно требования, а ветвями – домены. Корнем дерева является вершина, которая объединяет все домены, то есть непосредственно СТ. Пример реализации древовидной структуры представлен на рис. 1.

Целью иерархического представления требований является их структуризация. Структуризация ускоряет процесс разработки требований, позволяет выявлять неточности, ошибки и прочие проблемы еще на этапе анализа требований, кроме того, в дальнейшем упрощает процесс управления требованиями.

На предлагаемую модель накладываются следующие ограничения. Ребра взвешенного графа определяют принадлежность конкретного требования определенному домену. Вес ребер между ветвями и листьями равен единице, т.к. их отношения однозначно определены принадлежностью требования домену. Минимальная степень графа $deg(G)$ должна быть равна единице, т.к. не должно существовать изолированных вершин, другими словами, каждое требование должно быть отнесено хотя бы к одному домену. Диаметр графа $diam(G)$ не может быть больше 4. Это ограничение выражает тот факт, что домен не может содержать поддоменов:

если домен можно разбить на поддомены, тогда каждый поддомен должен быть охарактеризован как самостоятельный домен. Также и требование не может иметь дочерние требования. На данном этапе построения графа листья графа должны иметь только одно инцидентное ребро. Если лист имеет несколько смежных вершин-доменов, следует проанализировать данное требование и разбить его на несколько вершин таким образом, чтобы выполнялось наложенное ограничение. Все ограничения накладывались на модель в соответствии с теорией графов [7].

Следующий этап построения графовой модели – это задание множества характеристик требования в виде семантической аннотации.

Опишем требования, представленные текстом на естественном языке, ограниченным набором свойств малой длины, то есть каждое свойство должно быть минимально возможным количеством слов. Каждое требование обязательно должно иметь следующие свойства: объект, субъект, действие, условие выполнения. Определение для каждого требования всех четырех свойств, перечисленных выше, служит залогом обеспечения такого критерия качества как полнота второго типа. Таким образом, множество требований может быть представлено через множество свойств в этом частном случае в следующем виде:

$$R = \{p_i \mid p = \{O_i, S_i, A_i, C_i\}, i = \overline{1, n}\},$$

где p_i – множество свойств, в котором O_i – совокупность объектов, над которыми субъекты S_i совершают действия A_i при выполнении условий C_i .

В общем случае, множество требований может быть представлено в следующем виде:

$$R = \{p_i \mid p = \{P_j, j = \overline{1, m}\}, i = \overline{1, n}\},$$

где m – количество свойств, описывающих требование, P_j – j -ое свойство i -го требования.

Если требование имеет больше одного значения какого-либо свойства, например, в требовании описаны два различных действия над одним объектом, следует разделить данное требование с тем, чтобы обеспечить его атомарность. Логично, что, имея атомарные требования, каждое требование может быть проверено на выполнимость. Ребра невыполнимых требований можно выставить вес, равный 0. Теперь и в дальнейшем ребра, имеющие нулевой вес, удаляются из графа, также удаляются изолированные листья графа.

Кроме того, для повышения качества спецификации требований предлагается квантифицировать требования, то есть определять количественные значения в требованиях. Очевидно, что не любое требование может быть квантифицировано, однако использование квантификации даже там, где выбираемые значения очевидны, исключает использование таких слов, как «несколько», «много», «быстро» и так далее, что препятствует возникновению двусмысленности и способствует улучшению проверяемости требований.

Следующий этап построения графа – это усиление его связности. Между всеми листьями графа проведем ребра. Вес ребра выражается различием двух требований.

Примем мерой различия двух требований семантическое расстояние, которое является показателем смыслового различия и является действительным числом в интервале от 0 до 1, где 1 – требования идентичны, 0 – требования совершенно не связаны. Исходными данными для вычисления являются характеристики, которыми аннотированы требования. Для вычисления семантического расстояния между двумя требованиями используем расстояние Хэмминга [8] в следующем виде:

$$L = \frac{\sum_{i=1}^n (a \mid a = 0 : P_{i_1} \diamond P_{i_2}, a = 1 : P_{i_1} = P_{i_2})}{N}, \quad (1)$$

где L – семантическое расстояние, a принимает значения 0, когда свойства двух требований не равны, или 1 в противном случае, $P_{i(1,2)}$ – i -ое свойство каждого из двух требований, N – число свойств требования.

Ребра с нулевым весом удаляются из графа. Чем больше связей между вершинами, тем сложнее допустить ошибку при последующем изменении требований. Смежность листьев служит показателем того, что в СТ следует указывать связь этих требований. Таким образом осуществляется соответствие требования такому критерию качества требования как

трассируемость. Вес ребра в таком случае служит вероятностью добавления связей между требованиями. Для исключения избыточных связей (т.е. для того, чтобы граф не был перегружен большим количеством ребер) каждому свойству требования можно добавлять вес. Тогда формула (1) может быть представлена в следующем виде:

$$L = \frac{\sum_{i=1}^n (a | a=0: P_{i_1} \diamond P_{i_2}, a=1: P_{i_1} = P_{i_2}) * w_i}{N}, \quad (2)$$

где w – вес i -го свойства каждого из двух требований.

Если найдется такое ребро, что его вес будет равен единице, то велика вероятность того, что смежные листья этого ребра являются дубликатами одного и того же требования. Такие требования следует дополнительно анализировать.

Полученная графовая модель представлена на рис. 1, где корнем дерева является вершина, называемая SRS (от английского Software Requirements Specification), объединяющая в себе все домены D_i , которые включают в себя листья – конкретные требования, помеченные как R_i . Ребра, инцидентные вершинам D_i , имеют вес, равный единице в соответствии с ограничениями, наложенными на модель. Ребра, соединяющие вершины R_i , имеют вес, рассчитанный по формуле (2).

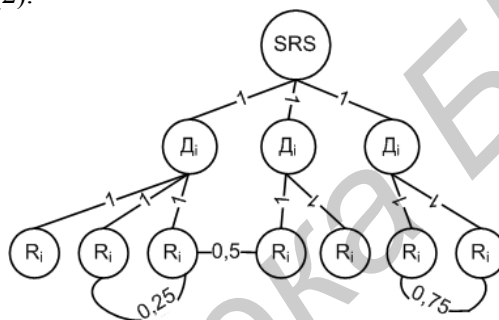


Рис. 1. Графовая модель представления спецификации требований

Приведем пример расчета весов ребер, соединяющих листья графа. Пусть существует два требования: «Непривилегированный пользователь может просматривать свои заказы», «Администратор может просматривать все заказы». Выделим свойства каждого из требований. Для первого требования объектом является заказ, субъектом – пользователь, действием – просмотр, условием – «только свои заказы». Для второго требования объектом является заказ, субъектом – администратор, действием – просмотр, условием – «все заказы». Пусть вес каждого свойства равен единице. Свойства «объект» и «действие» двух требований одинаковы, остальные различны. Таким образом, производя расчет по формуле (2), получаем вес ребра, равный 0,5.

Действия, выполняемые на каждом шаге построения предложенной модели СТ, в совокупности представляют собой метод, повышающий значения некоторых критериев качества, следовательно, повышающий качество спецификации требований.

Проблема определения противоречивых требований

Выявление противоречий в СТ является сложной задачей. Проблема заключается в том, что не только два атомарных требования могут противоречить друг другу, но и группа требований может противоречить одному требованию в то время, как по отдельности рассматриваемые требования не противоречат друг другу.

Рассмотрим случай, когда необходимо выявить два противоречивых требования. Требования являются противоречивыми по отношению друг к другу тогда и только тогда, когда они описывают действия над одним объектом, и хотя бы одна пара свойств другого типа, например, свойство «субъект», имеют значения, которые могут быть выражены в следующем виде: $P_{i_1} \Leftrightarrow \neg P_{i_2}$, т.е., когда значение свойства одного требования равносильно отрицанию значения того же свойства второго требования.

Таким образом, необходимо проанализировать все смежные листья графа. Если любая пара смежных листьев имеет противоречивые значения свойств одного типа, то вес их ребра становится отрицательным. Следует различать отличающиеся значения свойств от противоречивых значений. Отрицательное значение веса ребра указывает на то, что либо изначальные требования противоречивы, либо в СТ отсутствует связующее их требование, либо любое значение свойства противоречивых требований несправедливо.

Рассмотрим случай, когда необходимо выявить требование, противоречащее группе требований. Для этого удалим из графа все шарниры, являющиеся вершинами-доменами. Оставшиеся компоненты связности представляют собой наборы требований, связанные по каким-либо свойствам требования. Тщательному анализу должны быть подвержены циклы, так как именно в циклах вероятность возникновения противоречия выше.

На рис. 2 изображен граф, у которого удалены шарниры, то есть домены, которые связывают требования, а компоненты связности представлены в виде вершин (требований), соединенных дугами, и для наглядности обведены. Требования, которые не попали в компоненты, являются независимыми, следовательно, не должны вступать в противоречия с другими требованиями.

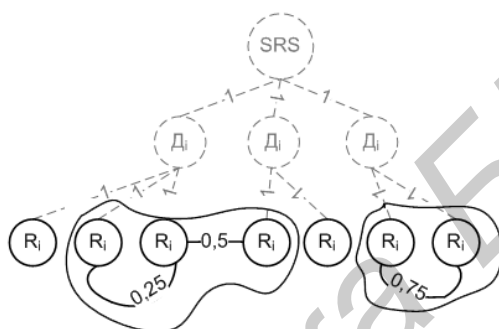


Рис. 2. Разбиение графа на компоненты путем удаления шарниров

Меры качества, основанные на предложенной графовой модели

Мера качества – производная мера, определяемая как результат функции измерения двух или более величин, входящих в нее элементов, определяемых в терминах свойств и методах измерения их количественных оценок [9]. Используя ограничения, наложенные на модель, а также свойства описанной модели, можно выделить следующие меры качества спецификации требований к программным средствам.

Данные меры можно использовать для оценки некоторых критериев качества спецификаций требований.

1. Полнота описания всех аспектов требования, включающих функциональные возможности, условия выполнения, ограничения проектирования, характеристики, внешние интерфейсы, реакции ПС на как допустимые, так и недопустимые значения входных данных во всех возможных ситуациях:

$$M_{compl} = \frac{\sum_{i=1}^n \sum_{j=1}^m P_{ij}}{nm},$$

где n – количество требований в спецификации, m – заданное количество характеристик каждого требования, P_{ij} – j -ое свойство i -го требования, которое в формуле принимает значение, равное единице, если свойство присутствует, и нуль в противном случае.

2. Показатель обеспечения трассируемости связанных требований:

$$M_{trace} = \frac{N_{traced}}{N_{related}},$$

где $N_{related}$ – количество всех связанных требований, N_{traced} – количество требований, для которых указана ссылка на связанное с ним требование.

Диапазон значений предложенных мер лежит в отрезке [0;1].

Экспериментальное подтверждение повышения качества при помощи предложенной модели и мер качества

Для проведения эксперимента были рассмотрены три спецификации требований к разным программным средствам, составленные профессиональными бизнес-аналитиками.

Требования были реструктурированы и аннотированы. Для каждой спецификации требований был построен свой граф, а при построении графа по предложенной модели некоторые требования были переформулированы. Также было посчитано общее количество требований до и после преобразований; количество некачественных требований, выявленных во время построения модели: дубликаты, неатомарные требования, противоречивые требования; первоначальные и измененные спецификации требований были оценены по предложенным мерам.

В таблице приведены результаты эксперимента. После применения предложенной модели количество некачественных требований уменьшилось: дубликаты были удалены, неатомарные требования были перефразированы и разбиты на более детальные требования. Также были выявлены противоречивые требования. Такие требования были проанализированы, а противоречия разрешены.

После применения модели было подсчитано количество связанных требований, однако, не все связанные требования были трассированы в рассматриваемых спецификациях. Сильно связанные требования были протрассированы. Также спецификации были оценены предложенными мерами качества до и после применения модели. Из таблицы видно, что после применения модели полученные значения мер увеличились, что говорит о том, что уровень качества СТ повысился.

Результаты эксперимента

№ п/п	Всего требований (до/после)	Дубликаты/ Неатомарные/ Противоречивые	M_{compl} , $m = 4$ (до/после)	Количество связанных требований	Трассируемые требования (до/после)	M_{trace} (до/после)
1	100/116	2/17/1	0,613/0,765	86	25/50	0,291/0,581
2	200/221	3/15/2	0,688/0,881	72	19/46	0,264/0,639
3	150/155	0/3/0	0,875/0,934	52	30/42	0,577/0,808

Заключение

Предложенная графовая модель представления требований к ПС может быть применена при выявлении, сборе и анализе требований. Использование модели упрощает документирование требований, а также способствует выявлению ошибок при дальнейшем управлении требованиями. Применяя предложенный метод на практике, аналитик получает дополнительную возможность повысить качество требований как на начальном этапе создания СТ, так и на протяжении всего жизненного цикла ПС при изменяющихся требованиях. Проведенный эксперимент доказал, что значения некоторых критериев качества спецификаций требований могут быть повышены при применении предложенной модели. Очевидно, что применение модели к уже созданным спецификациям требований дает хорошие результаты, более того, применение метода на начальных этапах создания СТ должно упрощать работу с требованиями и исключать появление требований с низким качеством, повышая тем самым качество СТ в целом.

QUALITY IMPROVING GRAPH MODEL AND METHOD FOR SOFTWARE REQUIREMENTS SPECIFICATION

A.U. CHIRKOVA, V.V. BAKHTIZIN

Abstract

The graph model for software requirements specification is proposed. Besides, there are possible actions of improving the quality characteristics of software requirements described. In total proposed model and actions represent a method of the quality improvement of the software requirement specification.

Список литературы

1. *Вигерс К.* Разработка требований к программному обеспечению. М., 2004.
2. *Kotonya G., Sommerville I.* Requirements Engineering. Processes and techniques. New York, 1998.
3. *Hull E., Jackson K., Dick J.* Requirements engineering. London, 2005.
4. ISO/IEC 9126-1:2001. Software engineering – Product quality – Part 1: Quality model.
5. 830-1998. IEEE Recommended Practice for Software Requirements Specifications. 1998.
6. ISO/IEC 25010:2011. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models.
7. *Оре О.* Теория графов. М., 1980.
8. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. М., 1986.
9. *Бахтин В.В., Глухова Л.А., Неборский С.Н.* Метрология, стандартизация и сертификация в информационных технологиях. Минск, 2013.