раздел **МАТЕМАТИКА**

УДК 681.3

СЕМАНТИЧЕСКОЕ И МНОГОАСПЕКТНОЕ МОДЕЛИРОВАНИЕ В УПРАВЛЕНИИ ТРЕБОВАНИЯМИ К МАТЕМАТИЧЕСКОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

© А. М. Сулейманова, Н. Н. Яковлев*

Уфимский государственный авиационный технический университет
Россия, Республика Башкортостан, 450000 г. Уфа, ул. К. Маркса, 12.
Тел./факс: +7 (347) 273 78 23.
E-mail: nikolay.iakovlev@gmail.com

Выгода от повторного использования требований в IT-производстве ставит задачу поиска требований, учитывающего семантику требований. Задачу предлагается решить за счет категоризированной семантической аннотации требований. Предлагается метод категоризованной семантической аннотации, учитывающий структурные особенности формулировки требований. Также учитывается многоаспектность данных. В качестве области применения рассматривается IT-проект по разработке и внедрению автоматизированной системы управления документооборотом вуза и его подразделений.

Ключевые слова: *Требование, семантическая аннотация, концепт, управление требованиями, аспект, модель данных, семантический поиск*

Введение

Математическое и программное обеспечение (далее МиПО) информационных систем разрабатывается, а впоследствии эволюционирует в соответствии с требованиями, предъявляемыми бизнесом. Каждая итерация развития МиПО представляет собой IT-проект, результатом которого является очередной релиз МиПО, и, следовательно, проистекает по всем общепринятым правилам проектного менеджмента.

В научно-исследовательской работе CHAOS [1], выполненной Standish Group, указано 10 факторов, которые не позволяют вовремя выпускать IT-проект, придерживаться установленного бюджета и предоставлять требуемую функциональность заказчику. Приведем три самых значимых из них:

- Недостаток данных от пользователей;
- Незаконченные требования и спецификации;
- Изменение требований и спецификаций.

Эти факторы неудач связаны с требованиями к работе и эффективное управление требованиями (УТ) может снизить долю неудач.

Требования являются одним из ключевых компонентов проекта в соответствии с международным стандартом PMBoK4 (Свод знаний по управлению проектами) [4]. Также и в соответствии с общепринятой методологией IBM Rational Unified Process разработка или эволюция программного обеспечения (ПО) должна сопровождаться УТ к ПО, и как следствие, управлением ответами или реакциями на эти требования. Аналогичное условие выдвигают отечественные стандарты на информационные технологии ГОСТ Р ИСО/МЭК 12207 и ГОСТ Р ИСО/МЭК 15288.

Принципиальное отличие жизненного цикла (ЖЦ) информационной системы от материальной в том, что продукт производства ИС, которым является МиПО, может применяться неограниченное количество раз в любой географической точке, доступной по сети передачи данных. В особенности это актуально для производства МиПО на основе сервисно-ориентированной архитектуры (СОА). Продукт производства материальной системы на стадии применения ограничен как количественно, так и в возможности перемещения.

Особенностью разработки МиПО автоматизированной системы управления документооборотом (далее АСУД) вуза и его подразделений является сбор большого массива требований по подразделениям несколькими аналитиками и реализация этих требований большой группой разработчиков и специалистов по внедрению. В такой ситуации повышается риск дублирования и противоречивости требований, а также неоднократной реализации одинаковых или схожих требований, а это – нерациональное распределение трудовых ресурсов данного IT-проекта.

Постановка цели исследования

Целью УТ при управлении одним проектом является предоставление возможности контроля целостности требований, т.е. избежание дублирования и противоречивости требований. Требования могут дублироваться и противоречить друг другу, если в проекте участвует более одного аналитика или проект имеет большую протяженность во времени (1 год или более). Также при управлении несколькими проектами целью УТ является повторное использование требования из одного проекта в одном или нескольких других проектах. Актуальность повторного использования обусловлена также ростом популярности СОА ПО, одним из принципов которой является повторное использование требований и моделей. Основными задачами при этом является произведение семантической аннотации (далее СА) требований и быстрый автоматизированный поиск требований, семантически близких к заданному, на основе имеющейся СА, а также оперативное представление данных о требованиях в виде отчетов.

Решение этих задач выводит на качественно новый уровень процедуры работы с данными о требованиях, поэтому в этой работе также описана

* автор, ответственный за переписку

задача пересмотра модели данных требования в IT-проекте, которая имеет различные аспекты. Предлагается подход к ее решению в виде многоаспектной логической модели данных и метаданных.

Постановка проблемы исследования

Основная проблема заключается в том, что каждое требование представляет собой текст на естественном языке, осознание семантики которого требует определенных усилий даже для человека. При этом требование имеет также несемантические атрибуты, представляющие интерес для команды проекта: трудоемкость, комментарий архитектора, статус, ссылка на уже имеющееся решение и т.п.

В то же время, требования на МиПО системы среднего масштаба могут исчисляться сотнями, что делает практически невозможным мануальную работу с требованиями и предполагает необходимость автоматизации. Очевидно, без надлежащей формализации представления требований автоматизация невозможна.

К таким системам среднего масштаба относится АСУД вуза и его подразделений. В случае проекта по разработке такой системы в УГАТУ насчитывается около 50 бизнес-правил, 150 функциональных требований, а также около 100 требований технологического характера.

Рассмотрим ниже математическую постановку задачи. Домен – это группа проектов, объединенных единым семантическим пространством, т.е. пересечение множеств их концептов достаточно велико. Это могут быть проекты по доработке одного и того же исходного решения, либо проекты по автоматизации схожих процессов. Проект имеет множество требований. Каждое требование включает в себя текстовую формулировку, семантические данные, служащие координатными с точки зрения поиска, и атрибутивные данные. Все требования проекта (набор требований) – это модель системы, являющейся целью (планируемым результатом) проекта в терминах требований:

$D = \{d_1, d_2, ..., d_n\}$ – множество предметных областей (доменов), $n \in N$;

$Concepts_i$ – множество концептов i-того домена, где $i = \overline{1,n}$, с точки зрения векторного анализа является базисом;

$P_i = \{p_1^i, p_2^i, ..., p_m^i\}$ – множество проектов i-того домена, где $i = \overline{1,n}$;

$R_{ji} = \{r_1^{ji}, r_2^{ji}, ..., r_l^{ji}\}$ – множество требований j-того проекта i-того домена, где $j = \overline{1,m}$;

$r_k^{ji} = \{text^{kji}, C^{kji}, A^{kji}\}$ – вектор атрибутов k-того требования (где $k = \overline{1,l}$) j-того проекта i-того домена, где $text^{kji}$ – это текстовая формулировка требования, $C^{kji} = \{c_1^{kji}, c_2^{kji}, ..., c_q^{kji}\}$ множество концептов, а каждый концепт c_x^{kji} принадлежит суммарному множеству, т.е. $c_x^{kji} \in Concepts_i$, а $A^{kji} = \{a_1^{kji}, a_2^{kji}, ..., a_s^{kji}\}$ – множество несемантических атрибутов требования;

$\left|C^{kji} \cap C^{k'j'i}\right|$ – число элементов пересечения двух множеств концептов, определяющее сходство требований r_k^{ji} и $r_{k'}^{j'i}$: увеличение этого показателя говорит о более сильном сходстве. Следует обратить внимание, что литера i не имеет штриха – домен для обоих требований один и тот же. Задача аналитика: задав множество C^{kji} для исходного требования r_k^{ji}, найти одно или несколько требований $r_{k'}^{j'i}$, для которых $\left|C^{kji} \cap C^{k'j'i}\right| \to \max\left(\left|C^{kji}\right|, \left|C^{k'j'i}\right|\right)$ с целью получить доступ к атрибутам $A^{k'j'i}$.

Задачей настоящего исследования является разработка метода, позволяющего аналитику получить для требования r_k^{ji} множество концептов C^{kji} из его заголовка $text^{kji}$.

Анализ литературы

В работе [3] Сойонг Пак рассматривает подход к повторному использованию требований, однако не рассматривает методы их СА. В работе Юн Лин [5] рассматривает подходы к семантической аннотации процессных моделей и проектирует прототип программного средства для управления семантикой процессных моделей, однако не рассматривает УТ, представленными в виде текста на естественном языке. В работе [2] рассматривается комплексное средство использования знаний о проекте и УТ, но в ней рассматриваются требования не к МиПО в IT-проекте, а к инженерным сооружениям. Ни в одной из вышеприведенных работ не рассматривается СА требований, сформулированных на естественном языке, либо подходы, связанные с многоаспектностью данных.

Предлагаемый подход к семантическому поиску

Семантическая аннотация требований – это смена вербальной семиотической системы выражения требования на концептуальную семиотическую систему, у которой степень формализации семантики с точки зрения требования максимальна.

Предлагается описывать каждое требование с помощью конечного множества концептов, то есть производить их СА. Совокупное множество концептов всех требований в проекте или группе проектов также должно быть конечно. Предлагается метод СА требования, основанный на разделении концептов требования на категории с учетом специфику инженерии требований и ее методологий.

В соответствии с устоявшимися практиками требование всегда содержит такие категории как субъект, действие, объект и обстоятельство. В рамках определенной предметной области набор категорий можно расширить, например, для АСУД вузом можно добавить такие категории как «документ», «модуль», «адресат».

Концепты предлагается разбивать по этим категориям. В рамках одной предметной области

число категорий, а, следовательно, число аннотирующих концептов всегда будет одинаково, что:

1) позволит представлять аннотацию требования в виде вектора, и в качестве меры сходства требований использовать расстояние Хэмминга [6],

2) делает идентифицирующие семантические данные также и координатными,

3) позволяет строить многомерное пространство концептов, в котором каждое требование является точкой с заданными координатами (это может служить основанием для многомерного OLAP-анализа как еще одного средства повышения качества визуализации данных о требованиях).

Вектор требований – это по сути (n–1)-местный предикат, где n – число категорий требований, актуальных для данной предметной области. Категории образуют n-мерное пространство. В рамках одного проекта и всего домена число категорий постоянно.

Метод категоризованной СА вводит также множество категорий домена $Cat_i = \{cat_1^i, cat_2^i, ..., cat_{q_i}^i\}$, где q_i – число категорий в домене, тогда $\forall t = \overline{1,q} \Leftrightarrow c_t^{kji} \in V(Cat_t^i)$.

При использовании категоризованной СА с постоянным числом категорий в рамках одного домена для поиска схожих требований предлагается использовать расстояния Хэмминга. Предлагаемый алгоритм поиска семантически похожих требований включает следующие операции:

- в *j*-том проекте *i*-того домена вводится требование r_k^{ji};
- для требования r_k^{ji} задается множество концептов $C^{k'j'i} = \{c_1^{k'j'i}, c_2^{k'j'i}, ..., c_{q_i}^{k'j'i}\}$, по одному концепту на каждую из q_i категорий, где q_i – число категорий i-того домена;
- для $k = \overline{1,l}$ при $j = \overline{1,m}$ вычисляется расстояние Хэмминга $d(C^{k'j'i}, C^{kji}) \in [0; q_i]$ и если $d(C^{k'j'i}, C^{kji}) \geq d_0$, где d_0 – условный уровень значимости, заданный аналитиком, то утверждается, что r_k^{ji} семантически похоже на $r_{k'}^{j'i}$;
- если $j = j' \cup k \neq k'$, то возможно требование $r_{k'}^{j'i}$ дублирует либо противоречит требованию r_k^{ji};
- если $j \neq j'$, то, возможно, что его атрибуты A^{kji} представляют интерес при реализации $r_{k'}^{j'i}$.

Аннотация может производиться как вручную (вводом концепта с клавиатуры или выбором из списка уже имеющихся), так и автоматически (система обрабатывает текст требования с учетом морфологизации, и если находит в тексте требования слово, уже зафиксированное ранее как концепт этого домена, значение подставляется в соответствующее поле автоматически).

Таким образом, авторами предпринята попытка впервые представить метод СА для требования в IT-проекте на естественном языке, учитывающий структурную специфику требования и предусматривающий единообразную категоризацию аннотирующих концертов для всех требований в рамках проекта.

Предлагаемый подход к логической модели данных о требованиях

Обозначим множество атрибутов одного требования как *Attr(R)*, где *R* – это требование.

Тогда $A(F,U,S) = \{a_1; a_2; ...; a_n\}$, $n \in N$, где

F – область знаний (*field*, для данной работы перечень областей взят из свода знаний по проектному менеджменту исследовательского института PMI [4]);

U – роль пользователя (User);

S – стадия ЖЦ;

и при $i = \overline{1,n}$ $\forall a_i \in Attr(R)$

Таким образом, выделено три аспекта, один из которых определяет основание для визуальной группировки атрибутов, а два других – роль пользователя и стадия ЖЦ определяют, может ли текущий пользователь редактировать или вводить данный атрибут на данной стадии ЖЦ, что служит основанием для многоаспектного разграничения доступа к данным о требованиях и, следовательно, повышает уровень целостности данных.

В соответствии с этой метамоделью данных, построена многоаспектная логическая модель данных, фрагмент которой представлен ниже.

Таблица

Фрагмент многоаспектной логической модели (словаря) данных о требовании.

Атрибут	Аспект			Обязательность.	Тип	Многозначность
	Область	Стадия	Роль			
Описание	Объем	Выявление	Аналитик		Текст	
Заголовок	Объем	Докум-ние	Аналитик	Да	Текст	
Родительское требова-	Объем	Докум-ние	Аналитик		Требование	
Трудоемкость (план)		Валидация	Архитектор		Число	
Аналитик	Ресурсы	Докум-ние	Система		Категория	
Архитектор	Ресурсы	Валидация	Система		Категория	
Уровень риска	Риски	Валидация	Архитектор		Категория	
Классы	Интеграция	Разработка	Разработчик	Да	Категория	Да
Файлы	Интеграция	Разработка	Разработчик	Да	Категория	Да
Трудоемкость (факт)	Стоимость	Разработка	Разработчик		Число	
Дата завершения (план)	Время	Валидация	Архитектор		Дата	
Релиз	Интеграция	Все	Аналитик		Категория	

Перспективы исследования

Перспективной представляется Классификация требований, например, между требованиями на исправление ошибки и требованиями на разработку функциональности. Также определение трудоемкости (не точное, а на уровне интервала) выполнения требования. Задача классификации при условии семантической аннотированности требования представляется типичной задачей интеллектуального анализа данных (*Data Mining*), где нужно определить нужную категорию сущности в зависимости от ее дискретных либо количественных характеристик на основе прецедентов, т.е. требований, трудоемкость и другие характеристики которых уже установлены опытным путем (за счет реализации).

Заключение

Результатом реализации данного подхода стала система управления требованиями SemanticReq (semreq.anna-marry.ru), позволяющая вести учет требований в проектах, производить семантическую аннотацию требований с учетом их семантических категорий, а также осуществлять поиск требований на основе произведенной семантической аннотации.

Система используется в проектах, связанных с автоматизацией документооборота кафедр и деканата УГАТУ. В результате использования было выявлено 8 дубликатов и 2 противоречия внутри проектов, а также было найдено 5 требований, аналогичных уже реализованным, что позволило избежать дополнительных затрат на разработку.

ЛИТЕРАТУРА

1. CHAOS // The Standish Group Report, 1999. URL: http://www.cs.nmt.edu/~cs328/reading/Standish.pdf.
2. Baxter D., Gao J., Case K.. An engineering design knowledge reuse methodology using process modelling. URL: https://dspace.lib.cranfield.ac.uk/handle/1826/1856.
3. Soyong P. Software Requirement Text Reuse. URL: ftp://ftp.umcs.maine.edu/pub/WISR/wisr6/proceedings/ps/park2.ps.
4. Project Management Body of Knowledge v.4 // Project Management Institute, 2004 URL: www.pmi.org.
5. Yun Lin. Semantic Annotation of Process Models. Facilitating Process Knowledge Management via Semantic Interoperability // Thesis for the degree of Ph.D. 2008. URL: http://ntnu.diva-portal.org/smash/get/diva2:124135/FULLTEXT01.
6. Hamming R. W. (1950). Error detecting and error correcting codes // Bell System Technical Journal 29 (2) C. 147–160

Поступила в редакцию 11.06.2010 г.