

Н.В. ПУСТОВАЛОВА, Т.В. АВДЕЕНКО  
**ПОСТРОЕНИЕ СОГЛАСОВАННОЙ МОДЕЛИ ТРЕБОВАНИЙ  
ДЛЯ ПРОЦЕССА ПРОГРАММНОЙ ИНЖЕНЕРИИ**

---

*Пустовалова Н.В., Авдеенко Т.В. Построение согласованной модели требований для процесса программной инженерии.*

**Аннотация.** В работе описан подход, позволяющий сформировать согласованную модель типов требований для конкретного проекта по разработке программ. Использование такой модели призвано разрешить противоречия, вызванные разным пониманием определения «требование к программе», а также скоординировать различные модели типов требований, лежащие в основе отдельных видов документов-спецификаций и методологий работы с ними. Для построения согласованной модели исследованы наиболее часто используемые виды документов-спецификаций и модели требований, лежащие в их основе. В качестве базиса для построения классификации типов требований использована теория полевой структуры частей речи. Сформулировано определение «тип требования». Для этого выделен ряд критериев, пригодных для оценки отдельных экземпляров требований на их принадлежность к типам и полям требований. Такой подход позволяет на основании набора признаков экземпляра требования отнести его к определенному типу, а затем рекомендовать включить в структуру спецификации типы требований, необходимые для обеспечения качества спецификации.

**Ключевые слова:** программная инженерия, инженерия требований, спецификация требований, непротиворечивость и полнота требований, теория полевой структуры частей речи, онтология.

*Pustovalova N.V., Avdeenko T.V. Building a Harmonized Model of Requirements for Software Development Process.*

**Abstract.** The paper describes an approach to form a harmonized model of requirements for a specific software development project. Such a model is intended to resolve the contradictions caused by different understanding of the "program requirements" definition, as well as to coordinate the various models of requirements types that underlie certain types of documents, specifications and methodologies of requirements engineering. The most commonly used examples of requirements specifications were analyzed. We propose the theory of the field structure of parts of speech as a basis for requirement type classification and give a special definition for "requirements types." In addition, we propose a set of criteria for identification of types and fields of requirements. Based on a set of criteria for a requirement instance, this approach allows one to identify the requirements type and then recommend adding desired types to the requirements specification.

**Keywords:** software engineering, requirements engineering, requirements specification, completeness and consistency of the requirements, field theory in linguistic and grammar, ontology.

---

**1. Введение.** Прошло более 30 лет с момента опубликования эссе «No Silver Bullet» [1], в котором были рассмотрены различные проблемы, обозначившиеся после того, как стало ясно, что разработка программного обеспечения (ПО) превращается в массовый технологический процесс. Основным источником проблем, «оборотнем», который, неожиданно преобразаясь, губит весь проект, были названы требования к программам. Именно после этого инженерия требова-

ний (Requirements Engineering, RE) начала активно развиваться как самостоятельная область знания. Несмотря на все усилия специалистов, направленные на формализацию и стандартизацию процесса работы с требованиями, он по-прежнему во многом носит творческий и случайный характер. Это порождает ряд проблем в процессе программной инженерии: срыв сроков, превышение бюджета, создание некачественного программного продукта. В качестве одной из причин можно назвать отсутствие единой и согласованной теоретической базы инженерии требований [2, 3]. В настоящее время при работе с требованиями используется несколько различных подходов, в основе которых лежат разные классификации типов требований.

Сегодня существует множество типов разрабатываемых программ, специфику которых необходимо учитывать при организации процесса программной инженерии. Создание и внедрение формализованных процессов разработки, адаптированных для типовых проектов, будет полезно с точки зрения сокращения затрат, а также повышения уровня зрелости процессов организации-разработчика согласно модели CMM (Capability Maturity Model).

В модели жизненного цикла ПО этап сбора и анализа требований предшествует остальным этапам. Результат работы с требованиями записывается в виде спецификации (Software Requirements Specification, SRS). По определению [5] спецификация требований — это документ, который полно и четко определяет системные, программные, функциональные требования к ПО и его компонентам, параметры, которым они должны соответствовать, определяет дальнейшую процедуру проведения верификации и валидации. Спецификация требований является основой для всего процесса разработки, ее качество напрямую влияет на качество итогового продукта. Определение параметров спецификации, оказывающих значимое влияние на качество итогового продукта с учетом характеристик реализуемых проектов [4], позволит формировать рекомендации по созданию итогового документа-спецификации (технического задания), учитывающего особенности конкретного проекта.

Спецификация требований используется на различных этапах процесса разработки программ: проектирование и кодирование программ, подготовка и проведение тестирования, аттестации ПО. Очевидно, насколько важно корректно сформулировать требования и отразить их в виде спецификации, позволяющей обеспечить реализацию основных свойств требований: адекватность, полноту, однозначность, непротиворечивость, трассируемость [2, 6, 7]. Адекватность системы требований заключается в том, что на ее основе может быть разрабо-

тана программа, выполняющая именно те функции, которые нужны заказчику. Свойство полноты обеспечивается, когда требование содержит всю необходимую информацию для своей корректной реализации. Однозначность заключается в том, что каждая формулировка конкретного требования может быть интерпретирована единственным образом. Непротиворечивость подразумевает, что одно конкретное требования не должно содержать сведения, противоречащие другому. Трассируемость подразумевает, что всегда можно определить, с какими другими требованиями связано текущее, а также как оно реализуется в рамках этапов проектирования, кодирования и тестирования. Выполнение этих свойств определяет качество спецификации требований. Таким образом, построение качественной спецификации является основной целью инженерии требований.

Кроме сложности, связанных непосредственно с написанием и использованием текста спецификации, процесс инженерии требований также имеет особенности организационного характера. В [8] особо отмечается, что продуктивная работа с требованиями возможна только в рамках слаженной команды разработчиков, а для этого, безусловно, необходимо единообразное или согласованное представление об организации этого процесса. Кроме того, разработка больших систем требует декомпозиции, и выполнения работ по созданию отдельных подсистем разными коллективами. Взаимопонимание, общение на одном языке в рамках единой методологии может внести большой положительный вклад в успех всего проекта, упростив процесс выявления и управления требованиями [2].

Общепризнанным средством решения слабо формализуемых задач, к каким относится инженерия требований, являются методы, развиваемые в научной области искусственного интеллекта. В работе [9] было предложено описание формальной гибридной модели требований, включающей фреймовую онтологию и гибридный механизм вывода на знаниях. В работе [10] мы обосновали использование логической модели представления знаний для обеспечения выполнения таких свойств системы требований, как, например, непротиворечивость и полнота. В настоящей работе данная модель адаптируется и расширяется для решения проблемы согласования теоретической базы инженерии требований. В рамках модели определяются различные типы требований и устанавливаются иерархические и ассоциативные отношения между ними. На основе критического анализа существующих моделей требований строится онтология типов требований, определяющая отношения между типами требований (классами онтологии), с использованием теории полевой структуры частей речи.

Построенная онтологическая модель может служить основанием для последующего соотнесения тех или иных артефактов инженерии требований, что, в свою очередь, позволит определить сочетаемость и возможность совместного использования различных подходов, инструментов, методов, стандартов, либо конвертацию артефактов требований из одного вида в другой. Кроме того, имея общее представление о соотнесении типов требований, можно формировать рекомендации по включению тех или иных типов требований в структуру спецификации для обеспечения свойств требований.

**2. Анализ существующих моделей требований и попытка их унификации.** Структура документа-спецификации, описывающего требования, набор используемых в нем типов требований и их отношения зависят от выбранной для проекта модели требований. Ее выбор определяется используемой моделью жизненного цикла ПО, зрелостью процессов согласно СММ, навыков разработчиков, требований к стандартизации программного продукта и процессу его создания со стороны заказчика. Часто использование только одной модели с ограниченным набором типов требований недостаточно для выполнения свойств требований, а значит, обеспечения качества спецификации. По этой причине в процессе разработки приходится решать дополнительные вопросы, связанные гармонизацией артефактов, созданных не только с привлечением элементов нескольких моделей, но и в виде разных видов документов-спецификаций, возможно, разными коллективами разработчиков. На сегодняшний день разные модели основываются на двух подходах к классификации типов требований, рисунок 1 [3, 8, 11, 12]. При первом подходе все требования делят на функциональные и нефункциональные, как представлено на рисунке 1А.

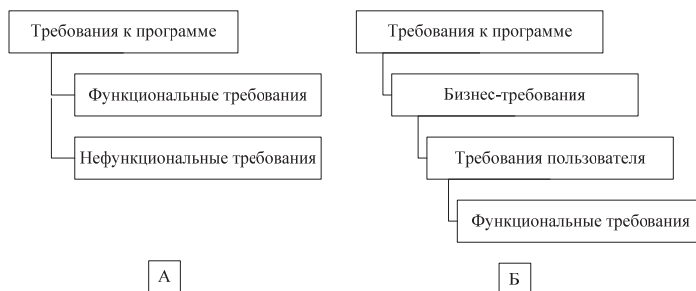


Рис. 1. Классификационные основания типов требований

В работе [3], функциональные требования определены как взаимодействие компонентов и их окружения, а нефункциональные как ограничения решений, которые могут быть приняты. В основе второго

подхода разделение требований по уровню детализации, представленное на рисунке 1Б [2, 7, 12, 14]. В рамках этого подхода выделяют: бизнес-требования, требования пользователя, функциональные требования. Стоит подчеркнуть, что для моделей, относящихся к этому подходу, понимание функциональных требований не полностью соответствует пониманию аналогичного термина при первом подходе. Редко для этого типа используется синоним «behavioral requirements», или требования поведения, как это происходит при первом подходе. Но в любом случае, его назначение - описать, что разработчику необходимо реализовать. В этом случае тип «функциональные требования», используемый в моделях, основанных на втором классификационном подходе, является частным случаем (подтипом), того, что используется в моделях, базирующихся на подходе первом. Независимо от того, какое основание было положено в основу модели типов требований, далее разработчики идут по пути детализации, дополняя ее описанием подтипов. Конкретные наборы подтипов и отношения между ними определяются методологией разработки, и составляют основу структур стандартов, на основании которых выполняются экземпляры спецификаций [7, 11–13]. В качестве примера можно привести выделение в структуре нефункциональных требований таких подтипов как «Требования к надежности», «Требования к составу и параметрам технических средств» согласно [11], «Требования к безопасности», «Требования к эргономике и технической эстетике» согласно [12], или атрибутов качества [5].

Если типы требований выделяются на основании критерия «функциональности», не происходит разделения спецификации по уровням детализируемых объектов, не в полной мере реализованы аспекты трассировки и наследования требований. В [8] приводятся данные о том, что описание иерархии требований в разрезе иерархии «родитель — потомок» улучшает качество спецификации и облегчает работу с ней. В случае выделения типов требований на основании разделения по уровням владельцев, отсутствует группировка по типам внутри уровней декомпозиции т.е. требования не разделяются на группы различных типов. Для небольших систем такой вариант структуры спецификации может быть допустим, но при разработке больших проектов такого разделения будет недостаточно для организации трассировки. Дополнительная декомпозиция может быть полезна, когда дело касается управления проектом, распределения функций между исполнителями, а также создания системы «зрелых» процессов инженерии программ. Использование одних и тех же названий для определения разных типов требований, и разное понимание места типов в структуре спецификации затрудняет процесс работы с требованиями, взаимодействие аналитиков.

На основе двух классификационных оснований (рисунок 1) формируются все остальные модели типов требований, задающие структуры итоговых документов-спецификаций. Эти документы задают структуру описания, его формат, но в вопросах наполнения оставляют аналитику достаточно много свободы. Причем даже в рамках одной методологии разработки существуют разногласия по поводу того, как использовать те или иные нотации для создания артефактов (например, можно взглянуть на UML диаграммы, создаваемые разными аналитиками). Отчасти это объясняется тем, что сами методики создания артефактов имеют рекомендательный характер и зависят от объекта предметной области, для моделирования которого создается артефакт, а также от принятых в организации практик их использования. «Одна из проблем, существующих в индустрии программного обеспечения, — это отсутствие общепринятых определений терминов, которыми мы пользуемся для описания нашей работы» [2]. Необходимость упорядочить и определить вопросы, касающиеся терминологии и методологии предметной области инженерии требований все еще есть.

Карл Вигерс, известный специалист в области инженерии требований, автор многих публикаций, одним из первых открыто заявил о необходимости комбинирования разных типов требований в процессе формирования спецификации [2]. На рисунке 2 представлен результат такого комбинирования.

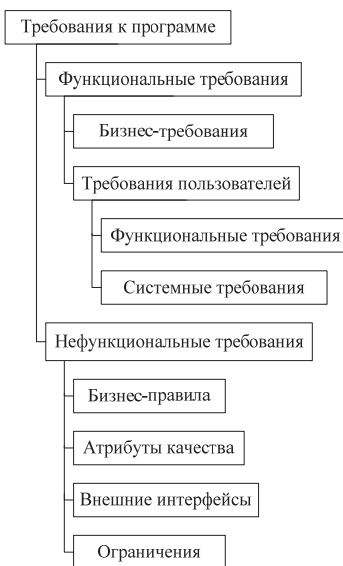


Рис. 2. Сводная модель типов требований, предложенная в [2]

Представленная модель содержит типы требований, ориентированные на описание как поведения, так и свойств программы, причем с учетом возможной детализации. Однако, как можно видеть на рисунке, модель включает всего 3 уровня детализации. Это объясняется тем, что она ориентирована на применение спецификации в формате SRS, дополняемой диаграммами прецедентов UML. В рамках этой модели хорошо визуализируется противоречие предметной области, касающееся определения функциональных требований. Есть и другие примеры подобных попыток сопоставления типов разных классификаций, например [14].

Модель, представленная на рисунке 2, также иллюстрирует разногласие авторов в определении места «ограничений». Под этим термином разные исследователи подразумевают пересекающиеся, но неодинаковые множества требований. Поэтому аналитик требований постоянно вынужден решать задачу построения корректной спецификации, в структуре которой предусмотрены типы отношений, характеризующие все аспекты, необходимые для обеспечения свойств спецификации и направленные на проверку разных видов связей между типами требований.

Рассмотрев предложенные модели соотнесения различных типов требований, можно заметить, что при их создании были использованы разные основания, и их применение ориентировано на проверку определенных связей между типами требований, то есть различных свойств спецификации. Согласованная модель типов требований, сочетающая различные подходы и разрешающая вопросы, касающиеся неоднозначного определения места типов в структуре предметной области, должна допускать задание разнообразных связей, отражающих отношения между типами требований.

Очевидно, что просто заменой названий в этом случае не обойтись, так как останутся смысловые различия. Поэтому необходимо сделать объединенную модель более дружелюбной к тем, кто придерживается разных точек зрения на предметную область. Сделать это можно, если нивелировать различия в формулировках, и ориентироваться на суть концептов, для описания которых используются понятия предметной области.

**3. Построение обобщенной модели типов требований к программам на основе теории полевой структуры частей речи.** Согласно теории полевой структуры частей речи, в каждой из них существуют единицы, обладающие полным набором признаков этой части речи. Такие единицы составляют ее ядро. Также существуют единицы, обладающие только частью таких признаков, но все же относящиеся к указанной части речи. Они составляют периферию поля.

Появление этой теории помогло решить проблему, связанную с определением принадлежности спорных лексических единиц к определенным частям речи [15, 16]. Адаптируя теорию полевой структуры к процессу инженерии требований, получим следующие понятия. Поле — это множество типов требований, определяемое набором признаков, значимых при описании определенного аспекта спецификации (поведение системы, место в структуре спецификации и т.д.). Ядро поля — типы требований, обладающие всеми признаками, определенными для описания конкретного поля. Периферия поля — типы требований, частично обладающие признаками, определенными для описания конкретного поля. В процессе адаптации теории полевой структуры для нужд программной инженерии, необходимо определить те признаки, которые являются значимыми для классификации требований. Отнесение типов требований к разным полям призвано решить задачу создания спецификации, для которой выполняется свойство полноты, за счет перечисления всех необходимых аспектов ее описания: поведения и характеристик, назначения системы в целом и нужд конкретных пользователей.

Определим тип требования как формальный набор элементов, призванных отразить атрибуты и формулировку некоторого множества экземпляров требований, имеющих определенную функциональную область применения, уровень детализации и место в структуре спецификации. Гипотеза исследования заключается в том, что на основании характеристик проекта по разработке программного обеспечения для гарантий качества итогового продукта, можно подобрать наиболее подходящую структуру спецификации (набор типов требований с учетом отношений между ними), и этим обеспечить выполнение свойств требований, т.е. качество спецификации.

При решении об отнесении типа требований к конкретному полю рассматривались следующие группы признаков:

1. Уровень детализации экземпляров конкретного типа. Содержит указания на владельца требований и исполнителя, ответственного за реализацию. При формировании спецификации такое описание типов направлено на обеспечение трассировки требований и управления ими.

2. Область применения экземпляров конкретного типа. Каждый экземпляр предназначен для отражения определенного аспекта системы (характеристики программы, функционал, интерфейс, данные). При формировании спецификации описание требований с этой точки зрения позволяет обеспечить ее полноту и трассируемость в разрезе всего процесса программной инженерии.

3. Рекомендации по формулированию экземпляров типа. Отражает группы ситуаций, которым в соответствие можно поставить шаб-



лоны конструкций на естественном языке, чтобы в дальнейшем рекомендовать их в качестве основы для формулирования экземпляров. Данная точка зрения на формулирование экземпляров требований направлена на достижение таких свойств спецификации, как ясность, адекватность и однозначность.

4. Место в структуре документа-спецификации, согласно методологии разработки. Отражает иерархические и иные связи, возможные между экземплярами требований в структуре спецификаций. Например, дублирование требований, или наличие противоречий между ними. При формировании спецификации описание требований с этой точки зрения призвано обеспечить её полноту и непротиворечивость.

В таблице 1 показано, как данные группы признаков были использованы при классификации типов требований на примере типа «функциональное требование» из версии стандарта SWEBOOK.

Таблица 1. Характеристика типа «функциональные требования», построенная по результатам изучения SWEBOOK

Признаки	Стандарт (методология)	Практические рекомендации
1	2	3
1. Уровень детализации описания экземпляров типа:		
1.1. Бизнес-уровень	1	0,7
1.2. Системный уровень	0,5	0,5
1.3. Пользовательский уровень	0	0
1.4. Уровень функций	0	0
1.5. Уровень реализации	0	0
2. Область применения экземпляров типа:		
2.1. Характеристики ПО	0	0
2.2. Функционал	1	1
2.3. Интерфейс	0	0
2.4. Данные	0,5	0,5
3. Рекомендации по формулированию экземпляров типа:		
3.1. Описание объекта в структуре требования	0,5	1
3.2. Описания действия над объектом в структуре требования	0,5	1
3.3. Описание параметров объекта в структуре требования	0	0
3.4. Описание процесса, связанного с объектом в структуре требования	0	0
3.5. Описание структур данных, связанных с объектом, процессом	0,5	0,5
4. Место в структуре документа-спецификации, согласно методологии разработки		
4.1. Наличие экземпляров-потомков	1	1
4.2. Наличие экземпляра-предка	0,5	1
4.3. Наличие горизонтальных связей между экземплярами.	1	1

В результате для формирования структуры спецификации предлагается использовать следующие поля:

1. Поведение;
2. Характеристики;
3. Высокоуровневое описание;
4. Детализованные требования;
5. Интеграция;
6. Данные.

Решение о принадлежности типа требований к определенному полю принимается на основании того, обладает ли тип требуемым набором признаков. То есть к полям относятся свойства типов. Для каждого типа требований, определяемого в разных идеологиях разработки [2, 8, 11, 13], были построены таблицы, содержащие данные о выраженности исследуемых признаков (как в таблице 1). Для этого используется шкала от «0» до «1», где «1» означает максимальную степень выраженности признака, а «0» — его отсутствие. Были учтены как общие рекомендации методик и стандартов, так и рекомендации специалистов, либо устоявшиеся приемы и методы описания конкретных типов. В таблице 1 этот факт отражен в столбцах 2 и 3. После чего были составлены итоговые таблицы, где были записаны итоговые значения признаков. Если значения признаков различались менее, чем на 0,5, бралось максимальное из них. Если значения признаков согласно стандартам и рекомендациям различались более, чем на 0,5, использовалось среднее значение. В случае если по итоговой сумме значений признаков один тип может быть отнесен к периферии разных полей, анализируются исходные таблицы, и на их основании принимается решение об отнесении типа к периферии конкретного поля. Тип относится к тому полю, где сумма признаков оказалась больше. При построении онтологии в Protégé в структуре класса «Требование» был предусмотрен специальный слот для записи имени альтернативного поля. Когда совпадение значений признаков разных типов составляет более 80%, они определяются как синонимы, и этот факт также отражается в специальном слоте класса «Требование». Данная ситуация возникает, когда разные методологии используют одно основание для классификации, или ссылаются друг на друга.

**4. Реализация структуры спецификации, включающей поля и типы требований.** Признаки, использующиеся для характеристики типов требований, при программной реализации положены в основу шаблонов, позволяющих описывать экземпляры требований. Использование шаблонов предполагает методическую помощь аналитику. Пример шаблона приведен на рисунке 3.

Функциональное требование

Выберите или введите задачу  
 Закрепление бортов за рейсом

Укажите объекты предметной области, с которыми связано требование ?  
 Рейс  
 Борт  
 Добавить объект

Описания функционального требования ?

Объект 1	Опишите связь между двумя объектами предметной области	Объект 2
Борт	Выполняет	Рейс
Борт	Закреплен за	Рейс

Добавить описание связи

Укажите необходимые условия для выполнения действия ?  
 В систему введены данные о рейсе  
 Добавить условие

Укажите плановый результат выполнения требования  
 В окне «выполнение рейса» в выпадающем списке доступны для выбора разные борта  
 Добавить результат

Общая формулировка требования

Каждому рейсу, доступному в справочнике системы, можно назначить борта, выполняющие его. В справочнике должны отображаться только те борта, которые доступны для вылета в ближайшие 2 часа.

Владелец требования  
 Агент регистрации

Сохранить

Рекомендовано добавить

Нефункциональные характеристики

Описание структур данных

Введите описание требования в поля шаблона...

Рис. 3. Шаблон типа «функциональные требования»

Шаблоны формируются превентивно, согласно рекомендациям наиболее широко распространенных методик инженерии требований. В шаблоны можно добавлять или исключать элементы, необходимые или лишние для описания экземпляров типа. Набор элементов, которые могут входить в состав шаблона, определен в таблице 1. Данный шаблон предназначен для описания типа «функциональные требования», синонимом которого является тип 6. Его синонимом является тип «поведенческие требования». В структуре шаблона отражено наличие конкретных значений таких фасетов как «уровень детализации» (пользовательский уровень), «область применения» (описание объекта в структуре требования). Такому экземпляру требования соответствует один экземпляр класса «функциональные требования», в слотах которого располагается описание действия над объектами, и два экземпляра класса «Объект», содержащие описание борта и рейса, а также связей между ними. Типы требований, применяемые при формировании спецификации, реализованы с помощью онтологии. Набор признаков (таблица 1) отражен в структуре класса «Требование», который определен как класс-родитель для всех остальных классов, задающих конкретные типы требований.

Для решения проблемы обеспечения выполнения свойств спецификации мы предлагаем использование гибридной модели знаний, включающей фреймовую онтологию, содержащую иерархическое описание типов (классов) требований и ассоциативных связей между ними, и систему правил, основанную на продукционной (логической) модели представления знаний. Формальное детальное описание предлагаемой гибридной модели приведено в [9].

$$Model = \langle O_F, F, P, M \rangle,$$

где  $O_F$  — фреймовая онтология, задающая классы (типы) требований и связи между ними,  $F$  — конечный набор фактов (сохраненных экземпляров требований для конкретной задачи),  $P$  — конечное множество правил, описывающих процедурный компонент системы,  $M$  — гибридный механизм вывода на знаниях. Реализацию выполнения свойств спецификации требований обеспечивает согласованное взаимодействие всех компонент построенной модели под управлением гибридного механизма  $M$ . В частности, реализацию свойств непротиворечивости и полноты спецификации обеспечивает механизм логического вывода на основе правила *Modus Ponens*, обладающий свойствами полноты и непротиворечивости. Реализацию механизма взаимодействия иерархических и ассоциативных отношений онтологии обеспечивает выполнение остальных свойств спецификации.

Основу онтологии составляет построенная в данной работе на основе полевой теории таксономия типов требований, определяющая иерархические связи между ними. Для построения таксономии был проведен анализ различных моделей типов требований [2, 3, 8, 11–13, 17]. При построении таксономии выделяются концепты, основные понятия предметной области, и записываются в качестве классов. Каждый класс содержит набор слотов, в которых могут храниться значения определенных типов, характеризующие конкретные экземпляры классов [18]. Между классами могут быть установлены стандартные отношения [19], например, наследование, или части и целого, в зависимости от возможностей инструмента построения онтологий [18]. Кроме того, можно определять непредусмотренные инструментом отношения, задавая в качестве значений слотов имена объектов других классов онтологии, если это необходимо. Также слоты классов могут быть использованы для хранения понятий синонимов, использующихся для описания концепта предметной области, как, например, в [20]. Таким образом, онтология позволяет описать предметную область инженерии требований, даже с учетом наличия в ней разных точек зрения, предполагающих рассмотрение большого количества близких, но не совсем одинаковых по смыслу терминов.

Совокупность полей охватывает все структурные аспекты спецификаций, применяемые в современных документах, используемых для описания требований. За полями закреплены типы требований на основе совпадения наборов признаков. Для определения принадлежности экземпляра к конкретному типу применяются шаблоны формулировок требований. Типы значений элементов шаблона соответствуют классам онтологии, предназначенным для описания предметной области. Конкретные значения элементов шаблона соответствуют экземплярам типов требований. При вводе в структуру шаблона нового требования, объекты, описанные в нем, и отношения между ними, добавляются в модель предметной области. Для её формирования предусмотрены класс «Объект» и соответствующие слоты в его описании. Между экземплярами класса «Объект» устанавливаются связи посредством задания значений слотов. Таким образом, на основании спецификации формируется модель предметной области, которую визуально может проверить аналитик. Каждый объект связан с конкретной формулировкой требования. Если при проверке модели предметной области в ней отсутствует объект, это является признаком неполноты спецификации. Безусловно, такой способ проверки полноты имеет ограничения в силу того, что на одной схеме нельзя отобразить слишком много объектов и связей. Но для проектов, где спецификация содержит не-

много экземпляров требований и описывает узкую предметную область, способ может быть применен. В качестве примеров такого рода проектов можно привести проекты, разрабатываемые на основе ограниченных спецификаций. В этом случае особенно важно понимать модель предметной области или задачи, так как документация к такого рода проектам заведомо не содержит полных данных.

Онтология, реализованная в среде Protégé, представлена на рисунке 4.

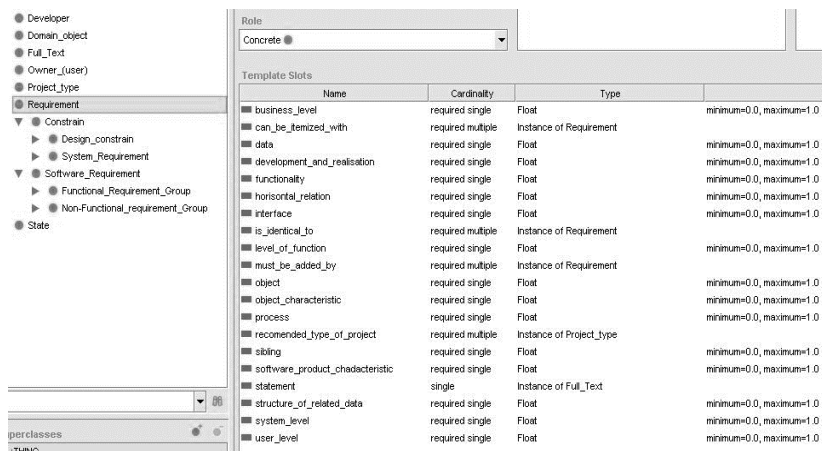


Рис. 4. Онтология понятий инженерии требований, реализованная в среде Protégé

Безусловно, применение онтологического подхода для записи требований при разработке узкоспециализированного ПО, имеет свои ограничения. В [21, 22] онтологии используются для формализации и анализа функциональных требований, хотя в [22] указывается на возможность использования онтологии пользовательских задач, для описания поведенческих аспектов будущей системы, то есть части нефункциональных требований.

В обеих представленных работах описание предметной области, используемое для формализации требований составлялось из нескольких онтологий, призванных отразить разные аспекты разрабатываемой системы: структуру, связи между элементами, характеристики и т.д. Развивая такой подход, необходимо предусмотреть онтологии, предназначенные для записи разных аспектов инженерии требований. В первую очередь, онтология типов требований, главная задача которой — отразить таксономию разных типов, с учетом их возможного дублирования и использования разных названий для описания одних и тех же кон-

цептов-классов. Для этого в данную онтологию необходимо добавить другие объекты предметной области инженерии требований, такие как функция, модуль и т.д. А также предусмотреть возможность хранения названий синонимов в слотах классов, описывающих понятия предметной области. Во-вторых, кроме онтологии инженерии требований, необходимо использовать онтологии пользовательских задач, где содержатся описания данных, поведенческой части системы и характеристик проекта. Это позволит соотнести виды спецификаций, представленные в онтологии инженерии требований, и типы проектов, что дает возможность на основании надстройки вывода рекомендовать тот или иной вид спецификации.

**5. Заключение.** В работе описан подход, позволяющий соотнести типы требований, относящиеся к разным классификациям и описывающие разные аспекты документа-спецификации. Его применение с учетом классификации проектов [4] позволяет формировать модели процессов программной инженерии, адаптированные для конкретных условий разработки, а также рекомендовать структуру спецификации, наиболее подходящую для конкретного проекта. Таким образом, аналитик требований и менеджер проекта получают методологическую поддержку, что способствует сокращению трудозатрат и обеспечивает дополнительный контроль качества в ходе разработки.

Совместное использование шаблонов и рекомендаций, касающихся структуры спецификации, призвано помочь аналитику учесть все аспекты описания требований. Таким образом реализуется свойство полноты. Реализация свойства непротиворечивости происходит посредством анализа связей объектов предметной области и требований, связанных с ними.

В работе [23] представлена формальная модель, описывающая взаимосвязь и реализацию таких свойств требований как полнота, целостность и непротиворечивость в процессе создания спецификации. Совместное использование данной модели и полевого подхода к описанию требований дает возможность для дальнейшей разработки методов, позволяющих осуществлять проверку свойств требований, а значит и качества спецификации, в автоматизированном режиме.

## Литература

1. *Brooks F.* No Silver Bullet: Essence and Accidents of Software Engineering // Computer. 1987. vol. 4. pp. 10–19.
2. *Wieggers K., Beatty J.* Software requirements // Microsoft Press. 2014. 637 p.
3. *Roman G.-C.* A Taxonomy of Current Issues in Requirements Engineering // IEEE Computer. 1985. pp. 14–21.
4. *Armin E., Li J.* Requirements Engineering: Technique Selection // In Encyclopedia of Software Engineering. 2011. pp. 962–978.

5. IEEE STD 610.12-1990. Standard Glossary of Software Engineering Terminology // NEW York: IEEE. 1990.
  6. *Кулямин В.В. и др.* Формализация требований на практике // М.: ИСП РАН. 2006. 50 с.
  7. IEEE Standard 830-1998. IEEE Recommended Practice for Software Requirements Specifications // NEW York: IEEE. 1998.
  8. *Leffingwell D., Widrig, D.* Managing Software Requirements - A Unified Approach // Addison-Wesley. 2000. 528 p.
  9. *Авдеенко Т. В., Бакаев М.А.* Гибридная модель представления знаний для реализации вывода во фреймовой онтологии // Научный вестник Новосибирского государственного технического университета. 2013. № 3. С. 84–90.
  10. *Avdeenko T.V., Pustovalova N.V., Klavsuts I.L.* The Ontology-Based Approach to Support the Requirements Engineering Process // In Proceedings of the 10 International Forum on Strategic Technology (IFOST-2015). Indonesia. 2015. pp. 70.
  11. ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению // М.: СТАНДАРТИНФОРМ. 2005.
  12. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы // М.: ИПК ИЗДАТЕЛЬСТВО СТАНДАРТОВ. 2004.
  13. *Орлик С.* Основы Программной Инженерии (по SWEBOK) // URL: [http://swebok.sorlik.ru/1\\_software\\_requirements.html](http://swebok.sorlik.ru/1_software_requirements.html). (дата обращения: 28.01.2014).
  14. *Cockburn A.* Writing Effective Use Cases // Addison-Wesley. 2001. 304 p.
  15. *Терпак М.А.* Фреймвая модель как способ структурирования и анализа концептов // URL: <http://vestnik-samgu.samsu.ru/gum/2006web51/yaz/2006510505.pdf>. (дата обращения: 28.01.2014).
  16. *Бондаренко А.В.* Проблемы функциональной грамматики. Полевые структуры // Спб.: Наука. 2005. 480 с.
  17. *Булуй Ю.И.* Классификация типов требований к программному обеспечению и ее представление в стандартах и методологиях // URL: [http://www.uml2.ru/index.php?option=com\\_remository&Itemid=28&func=fileinfo&id=135](http://www.uml2.ru/index.php?option=com_remository&Itemid=28&func=fileinfo&id=135) (дата обращения: 28.01.2014).
  18. *Noy N.F., McGuinness D.L.* Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880 // URL: [http://protege.stanford.edu/publications/ontology\\_development/ontology101.html](http://protege.stanford.edu/publications/ontology_development/ontology101.html) (дата обращения: 28.01.2014).
  19. *Найханова Л.В.* Основные типы семантических отношений между терминами предметной области // Известия высших учебных заведений. Поволжский регион. Технические науки. Пенза. 2008. № 1. С. 62–71.
  20. *Кортиков Ф.С.* Семантическое описание электронного документа с использованием онтологии предметной области // Труды СПИИРАН. 2013. № 8(31). С. 218–222.
  21. *Шалфеева Е.А.* Возможности использования онтологий при разработке и сопровождении программных систем // ИАПУ ДВО РАН. Владивосток. 2011. 16 с.
  22. *Шалфеева Е. А.* Модель связи структурных свойств онтологии со структурой модели требований системы, основанной на редактируемых знаниях // Искусств. интеллект. Донецк. 2009. № 4. С. 472–481.
  23. *Zowghi D., Gervasi V.* The Three Cs of Requirements: Consistency, Completeness, and Correctness // In Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02). 2002. pp. 155–164.
- 46 SPIIRAS Proceedings. 2016. Issue 1(44). ISSN 2078-9181 (print), ISSN 2078-9599 (online)  
[www.proceedings.spiiras.nw.ru](http://www.proceedings.spiiras.nw.ru)



## References

1. Brooks F. No Silver Bullet: Essence and Accidents of Software Engineering. *Computer*. 1987. vol. 4. pp. 10–19.
2. Wieggers K., Beatty J. Software requirements. *Microsoft Press*. 2014. 637 p.
3. Roman G.-C. A Taxonomy of Current Issues in Requirements Engineering. *IEEE Computer*. 1985. pp.14–21.
4. Armin E., Li J. Requirements Engineering: Technique Selection. In *Encyclopedia of Software Engineering*. 2011. pp. 962–978.
5. IEEE STD 610.12-1990. Standard Glossary of Software Engineering Terminology. NEW York: IEEE. 1990.
6. Kuliamin V.V., et al. *Formalizatsiia trebovanii na praktike*. [Requirements formalization in practice]. M.: ISP RAN. 2006. 50 p. (In Russ.).
7. IEEE STD 610.12-1990. Standard Glossary of Software Engineering Terminology. NEW York: IEEE. 1990.
8. Leffingwell D., Widrig, D. *Managing Software Requirements - A Unified Approach*. Addison-Wesley. 2000. 528 p.
9. Avdeenko T.V., Bakaev M.A. [Hybrid model of knowledge representation for inference realization in frame-based ontology]. *Nauchnyj vestnik Novosibirskogo gosudarstvennogo tehnikeskogo universiteta – Scientific bulletin of Novosibirsk state technical university*. 2013. vol. 3. pp. 84–90. (In Russ.).
10. Avdeenko T.V., Pustovalova N.V., Klavuts I.L. The Ontology-Based Approach to Support the Requirements Engineering Process. In *Proceedings of the 10 International Forum on Strategic Technology (IFOST-2015)*. Indonesia. 2015. pp. 70.
11. GOST 19.201-78. [Unified system for program documentation. Technical specifications for development. Requirements to contents and form of presentation]. M.: Standartinform. 2005. (In Russ.).
12. GOST 34.602-89. [Information technology. Set of standards for automated systems. Technical directions for developing of automated system]. M.: IPK IZDATEL"STVO STANDARTOV. 2004. (In Russ.).
13. Orlik C. *Osnovy Programmnnoi Inzhenerii (po SWEBOK)*. [Basic Software Engineering: SWEBOK]. Available at: [http://swebok.sorlik.ru/1\\_software\\_requirements.html](http://swebok.sorlik.ru/1_software_requirements.html). (accessed 28.02.2014). (In Russ.).
14. Cockburn A. *Writing Effective Use Cases*. Addison-Wesley. 2001. 304 p.
15. Terpak M.A. *Freimovaia model' kak sposob strukturirovaniia i analiza kontseptov*. [Frame model as a way to organize and analyze concepts]. Available at: <http://vestnik-samgu.samsu.ru/gum/2006web51/yaz/2006510505.pdf> (accessed 28.01.2014). (In Russ.). (In Russ.).
16. Bondarenko A.V. *Problemy funktsional'noi grammatiki. Polevye struktury* [Problems of functional grammar. Field structure]. St. Petersburg, Nauka. 2005. 480 p. (In Russ.).
17. Buluj Y.I. *Klassifikatsiia tipov trebovanii k programnomu obespecheniiu i ee predstavlenie v standartakh i metodologiiakh*. [Classification of requirements types and its representation as standards and techniques]. Available at: [http://www.uml2.ru/index.php?option=com\\_remository&Itemid=28&func=fileinfo&id=135](http://www.uml2.ru/index.php?option=com_remository&Itemid=28&func=fileinfo&id=135) (accessed 28.01.2014). (In Russ.). (In Russ.).
18. Noy N.F., McGuinness D.L. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. Available at: [http://protege.stanford.edu/publications/ontology\\_development/ontology101.html](http://protege.stanford.edu/publications/ontology_development/ontology101.html) (accessed 28.01.2014).
19. Naikhanova L.V. [The main types of semantic relationships between terms of domain area]. *Izvestija vysshih uchebnyh zavedenij. Povolzhskij region. Tehniceskie nauki –*

*Proceedings of the higher educational institutions. Volga region. Technical science.* Penza. 2008. vol. 1. 62–71 pp. (In Russ.).

20. Kortikov F.S. [The semantic description of electronic document using the domain ontology]. *Trudy SPIIRAN – SPIIRAS Proceedings*. 2013. vol. 8(31). pp. 218–222. (In Russ.).
21. Shalfeeva E.A. *Vozможnosti ispol'zovaniya ontologii pri razrabotke i soprovozhdenii programmyh sistem* [A possibility of using ontology during software development process and maintenance]. Vladivostok: IAPU DVO RAN. 2011. 16 p. (In Russ.).
22. Shalfeeva E. A. [Model of relationships between structural properties of ontology and structure of system requirements model that based on editing knowledge]. *Iskusstv. intellekt – Artificial intelligent*. Donetsk. 2009. vol. 4. pp. 472–481. (In Russ.).
23. Zowghi D., Gervasi V. The Three Cs of Requirements: Consistency, Completeness, and Correctness. In *Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02)*. 2002. pp. 155–164.

**Пустовалова Наталья Викторовна** — старший преподаватель кафедры экономической информатики факультета бизнеса, Новосибирский государственный технический университет (НГТУ). Область научных интересов: инженерия требований, программная инженерия. Число научных публикаций — 18. Natalja.ru@gmail.com, <http://ciu.nstu.ru/kaf/persons/22294/about>; пр. К. Маркса, 20 к.6, ауд. 601, Новосибирск, 630073; р.т.: +7(383) 346-0679.

**Pustovalova Natalya Victorovna** — senior lecturer of computer science in economics department of business faculty, Novosibirsk State Technical University (NSTU). Research interests: requirements engineering, software engineering. The number of publications — 18. Natalja.ru@gmail.com, <http://ciu.nstu.ru/kaf/persons/22294/about>; 20 k. 6, aud. 601, K. Marksa pr., Novosibirsk, 630073, Russia; office phone: +7(383) 346-0679.

**Авдеенко Татьяна Владимировна** — д-р техн. наук, профессор, заведующий кафедрой экономической информатики факультета бизнеса, Новосибирский государственный технический университет (НГТУ). Область научных интересов: интеллектуальные информационные системы, нечеткие вычисления. Число научных публикаций — 110. [tavdeenko@mail.ru](mailto:tavdeenko@mail.ru), <http://ciu.nstu.ru/kaf/persons/818>; пр. К. Маркса, 20 к.6, ауд. 601, Новосибирск, 630073; р.т.: +7(383) 346-0679.

**Avdeenko Tatyana Vladimirovna** — Ph.D., Dr. Sci., professor, head of computer science in economics department of business faculty, Novosibirsk State Technical University (NSTU). Research interests: intelligent information system, fuzzy computation. The number of publications — 110. [tavdeenko@mail.ru](mailto:tavdeenko@mail.ru), <http://ciu.nstu.ru/kaf/persons/818>; 20 k. 6, aud. 601, K. Marksa pr., Novosibirsk, 630073, Russia; office phone: +7(383) 346-0679.

## РЕФЕРАТ

### *Пустовалова Н.В., Авдеенко Т.В.* **Построение согласованной модели требований для процесса программной инженерии.**

В работе описан онтологический подход к формированию качественной спецификации требований, удовлетворяющей свойствам требований. Рассмотрены основные подходы к формированию спецификаций, типы требований, используемые в разных документах. На основании результатов анализа различных моделей требований предлагается использовать теорию полевой структуры частей речи для формирования согласованной модели требований, учитывающей все аспекты написания спецификации. Представлен набор критериев, используемых для определения принадлежности экземпляра требования к определенному типу, и типа к полю. Описан метод, которым построена онтологическая модель для решения задачи формирования спецификации требований, где обеспечиваются свойства требований.

Приводится краткое описание онтологии, применяемой для реализации полевого подхода в процессе инженерии требований. Приведен пример шаблона, предназначенного для работы с экземплярами требований в разрабатываемой среде.

## SUMMARY

### *Pustovalova N.V., Avdeenko T.V.* **Building a Harmonized Model of Requirements for Software Development Process.**

The paper describes an ontological approach to generate a high-quality requirements specification that meets the requirements properties. The main approaches to the requirement specifications development as well as different requirement types, used in different documents, are considered. Based on the results of the analysis of various requirement type models, we propose to use the field theory of parts of speech as a basis for a coherent model of requirements, taking into account all aspects of writing specifications. It presents a set of criteria used to identify the type and field for requirements. The paper presents a method of building an ontological model for the quality requirements specifications ensuring requirements properties.

A brief description of the ontology used to implement the field theory-based approach for requirements engineering process is given. An example of a software template for the requirement instance implementation is presented.