

АЛГОРИТМ ФОРМАЛИЗАЦИИ ТРЕБОВАНИЙ ПРИ РАЗРАБОТКЕ ВСТРАИВАЕМЫХ СИСТЕМ

Нахов В.А.

*Институт проблем точной механики и управления – обособленное структурное подразделение Федерального государственного бюджетного учреждения науки
Федерального исследовательского центра «Саратовский научный центр
Российской академии наук»;
АО «Конструкторское бюро промышленной автоматики», Саратов*

Ключевые слова: требования, алгоритм, системы, диаграмма деятельности, методы формализации требований, полнота требований.

Аннотация. Рассмотрен алгоритм формализации требований при разработке информационных систем. Выполнена доработка и адаптация алгоритма для определения требований к системам реального времени. На основе текстовых требований выявлены «актёры» и их действия. Разработана гибридная диаграмма деятельности. Исходя из полученных данных, проведена оценка алгоритма формализации требований.

AN ALGORITHM FOR FORMALIZING REQUIREMENTS IN THE DEVELOPMENT OF EMBEDDED SYSTEMS

Nakhov V.A.

*Institute of Precision Mechanics and Control – Subdivision of the Federal State
Budgetary Research Institution Saratov Federal Scientific Centre
of the Russian Academy of Sciences;
SC «Design Bureau of Industrial Automatics», Saratov*

Keywords: requirements, algorithm, systems, activity diagram, methods of requirements formalization, completeness of requirements.

Abstract. The algorithm of formalization of requirements in the development of information systems is considered. The algorithm has been finalized and adapted to determine the requirements for real-time systems. Based on the text requirements, the "actors" and their actions are identified. A hybrid activity diagram has been developed. Based on the data obtained, an evaluation of the requirements formalization algorithm was carried out.

Информационные технологии стремительно развиваются, а требования к программным продуктам становятся все более сложными и многообразными, и заказчики не всегда могут оценить сложность системы. Задача разработчика на этом очень раннем этапе проекта – выявить как можно больше требований, и определить язык написания этих требований который был бы понятен всем участникам работы [1]. Требования обычно записываются на естественном языке либо потому, что инженеры-разработчики не владеют формальными языками, либо потому, что на данном этапе разработки ещё слишком рано использовать такую форму спецификации. Основная проблема при написании требований на естественном языке заключается в том, что они не могут служить входными данными для методов автоматизированной проверки. Действительно, для применения методов автоматизированной проверки необходимы формальные

языки, поскольку это единственные языки, которые могут понимать компьютеры. Более того, формальные языки недвусмысленны: это означает, что предложение нельзя понять по-разному [2]. Поэтому формализация требований и оценка их полноты приобретают особую актуальность. Неправильное понимание или отсутствие четко сформулированных требований может привести к серьезным последствиям.

Большинство методов используют для формального описания готовые шаблоны [1, 3], или через использование графов [4].

В работе рассмотрен простой для пользователей алгоритм преобразования требований в формальную модель [5].

Для начала необходимо произвести синтаксический разбор каждого отдельно взятого требования и составить таблицу. Таблица должна содержать следующий набор столбцов подлежащее, сказуемое, дополнение и словосочетание в конце предложения.

Ввиду того, что текст требования по сути будет преобразован до формата: <Объект> <Действие> <Описание действия>, тогда от «Словосочетания в начале предложения», как указано в работе [5] можно отказаться.

Для проверки алгоритма преобразования требований в формальную модель будет использован набор требований для спутниковой системы CubeSat, и с помощью алгоритмов преобразования будет осуществлен переход небольшой части текстовых требований.

Исходный текст требований имеет следующий вид.

Оператор проводит подготовку CubeSat для установки в P-POD. Оператор подключает к пульту и подаёт наземное питание на CubeSat. С помощью пульта запускает проверку внутренних систем CubeSat. CubeSat запускает запись данных на бортовой регистратор. CubeSat проверяет внутренние системы и формирует признак исправности. Оператор осуществляет контроль процесса проверки внутренних систем CubeSat. Если проверка прошла успешно, оператор отключает наземное питание и переключается на автономное питание. Оператор открывает крышку люка для установки CubeSat. Оператор производит установку CubeSat в P-POD. В процессе установки оператор контролирует наличие отсутствия сигнала об ошибке установки. P-POD формирует сигнал о правильности установки CubeSat. Оператор закрывает люк. P-POD формирует сообщение о готовности. Далее, согласно алгоритму, описанные выше требования необходимо записать в виде табличного представления. Стоит отметить, что на данном этапе уже начинает наблюдаться неполнота требований. Далее заполняются в пустые ячейки таблицы, и таблица приобретает законченный вид (табл. 1).

По таблице 1 определяем актеров, т.е. действующих лиц (ДЛ): оператор, CubeSat, P-POD. Актеры выполняют следующие действия:

- оператор: делает, устанавливает, запускает, подключает, контролирует, отключает, открывает, закрывает;
- CubeSat: запускает, делает, формирует, переключает;
- P-POD: открывает, формирует, закрывает.

На последнем шаге алгоритма, путем использования простых правил преобразования, получаем диаграмму деятельности (рис. 1).

Табл. 1. Синтаксический разбор требований

№	Подлежащее	Сказуемое	Дополнение	Словосочетание в конце
1.	Оператор	делает	подготовку CubeSat	для
2.	ОПЕРАТОР	устанавливает	CubeSat в P-POD	
3.	Оператор	подключает	к пульту	
4.	ОПЕРАТОР	подаёт	питание	
5.	Оператор	запускает	проверку систем	
6.	CubeSat	запускает	запись данных	
7.	CubeSat	делает	проверку внутренние системы	и
8.	CUBESAT	формирует	признак исправности	
9.	Оператор	контролирует	проверку систем	
10.	CubeSat	формирует	признак неисправности	
11.	Оператор	отключает	наземное питание	и
12.	CUBESAT	переключает	на автономное питание	
13.	Оператор	открывает	крышку люка	
14.	ОПЕРАТОР	устанавливает	CubeSat	
15.	P-POD	открывает	крышку	
16.	Оператор	устанавливает	установку CubeSat	
17.	Оператор	контролирует	отсутствие сигнала об ошибке	
18.	Оператор	закрывает	крышку люка	
19.	P-POD	закрывает	крышку	
20.	P-POD	формирует	сообщение о готовности	

Каждое предложение в табличном представлении фактически – это элемент работы системы. Табличное представление можно рассматривать как базу знаний. Предложения упорядочены в таблицу, которая имеет все преимущества для обработки. Графическое представление является очень важным, поскольку позволяет визуально, быстро и точно определяется функции каждого из элементов системы. Также полученный граф может быть преобразован путем несложных изменений в гибридную диаграмму деятельности, из которой можно получить другие виды SysML-моделей [6, 7]. Алгоритм преобразования текстовых требований в гибридную диаграмму деятельности может быть автоматизирован, поскольку написан в виде псевдокода.

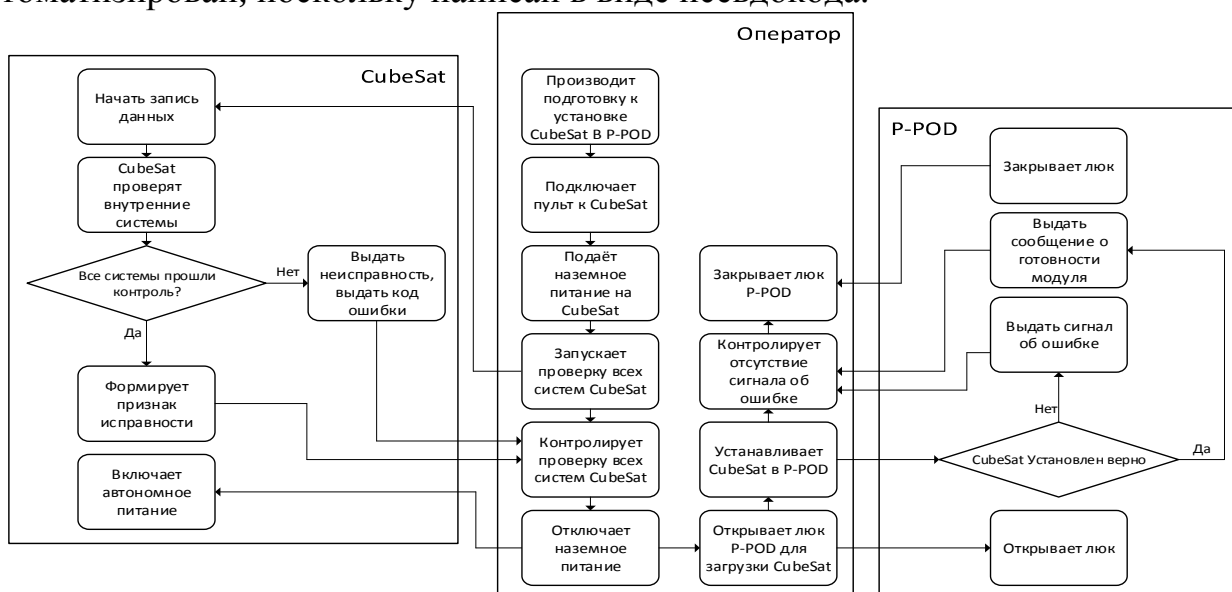


Рис. 1. Диаграмма деятельности

Рассмотренный алгоритм позволяет: оценить полноту требований и внести уточнения; получаемая формальная модель является полной, поскольку одновременно описывает как структуру и поведение системы. Гибридная диаграмма деятельности также включает понятия актеров, бизнес-правил и сообщений; есть возможность преобразовать созданную формальную модель в ряд других моделей SysML. Графическое представление позволяет использовать принцип модульности при дальнейшем проектировании информационной системы.

Финансирование. Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема №122030400209-9 Разработка интеллектуальных моделей и методов управления сложными человеко-машинными системами в условиях критических ситуаций).

Список литературы

1. Kuchta J. Completeness and Consistency of the System Requirement Specification // Federated Conference on Computer Science and Information Systems. 2016, no. 9, pp. 265-269.
2. Peres F., Yang J., Ghazel M. A Formal Framework for the Formalization of Informal Requirements // The International Journal of Soft Computing and Software Engineering. 2012, vol. 2, no. 8, pp. 14-27.
3. Халл Э., Джексон К., Дик Д. Разработка и управление требованиями. – Springer, London Berlin, 2005. – 226 с.
4. Bendík J. Consistency Checking in Requirements Analysis // International Symposium on Software Testing and Analysis. 2017, no. 26, pp. 408-411.
5. Липко Ю.Ю. Алгоритм формализации требований при разработке информационных систем // Известия ЮФУ. Технические науки. – 2014. – № 6(155). – С. 153-158.
6. Петров Д.Ю. Модельно-управляемое проектирование автоматизированной системы идентификации дефектов листового стекла // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2023. – № 4. – С. 41-48. – doi.org/10.24143/2072-9502-2023-4-41-48.
7. Мешалкин В.П., Чистякова Т.Б., Петров Д.Ю. Инжиниринг цифрового тренажера для обучения операторов формования листового стекла // Программные продукты и системы. – 2024. – Т. 37, № 2. – С. 561-566.

Сведения об авторе:

Нахов Владимир Андреевич – аспирант, инженер-системотехник второй категории.