

УДК 681.518

Ю.Ю. Липко**АЛГОРИТМ ФОРМАЛИЗАЦИИ ТРЕБОВАНИЙ ПРИ РАЗРАБОТКЕ
ИНФОРМАЦИОННЫХ СИСТЕМ***

Одной из актуальных проблем на начальных этапах жизненного цикла разработки информационных систем является формирование требований к системе, корректно и точно отражающих цели и задачи заказчика. Для успешной реализации проекта разработки информационной системы, отвечающей целям и задачам заказчика, нужно выяснить требования заказчиков к системе и преобразовать их на язык формальных моделей так, чтобы обеспечить соответствие целям и задачам организации. Для этого необходимо использовать наиболее эффективные методы преобразования требований в формальные модели. Объектом исследования являются подходы и методы преобразования требований в формальную модель при разработке информационных систем. Предложен алгоритм преобразования требований на естественном языке в формальную модель в виде гибридной диаграммы деятельности. Предложенный алгоритм позволяет получать основные и предметные знания из текстов требований, описывать структуру и поведение системы, преобразовывать созданную формальную модель в ряд других моделей UML, также предложенное табличное представление текста можно рассматривать как базу знаний при проектировании информационной системы. Практическая ценность результатов исследований определена их применением для решения проблем формализации требований на первом этапе жизненного цикла разработки информационных систем.

Технологии разработки информационных систем; преобразование требований; формальная модель; методы преобразования требований.

Yu.Yu. Lipko**THE ALGORITHM OF FORMALIZATION OF REQUIREMENTS WHEN
DEVELOPING INFORMATION SYSTEMS**

One of the urgent problems at the initial stages of life cycle information systems development is formation of system requirements, correctly and accurately reflect the goals and tasks of the customer. Organization of work with the requirements is among the priorities of the project management for the development of the information system, since the quality of the developed system is largely determined by how it meets the requirements. For the successful realization of the project of development of an information system that meets the goals and objectives of the customer, need to find out the requirements to the system and translate them into the language of formal models to ensure compliance with the goals and objectives of the organization. This requires the use of the most effective methods to transform requirements into formal models. The object of study are the approaches and methods for translating requirements into formal model in the development of information systems. In the work the algorithm of conversion requirements in natural language in a formal model in the form of hybrid chart activity. The proposed algorithm allows to obtain basic and subject knowledge of the texts requirements, to describe the structure and behavior of the system, convert created a formal model in a number of other UML model, as proposed tabular representation of the text can be considered as the knowledge base when designing information systems. Practical value of results of studies identified their application for solving problems of formalization of requirements in the first phase of the development life cycle information systems.

Technologies of information systems development; conversion requirements; formal model; methods of transformation requirements.

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 14-07-00910.

Для успешной реализации проекта разработки информационной системы, отвечающей целям и задачам заказчика, нужно преобразовать требования на язык формальных моделей так, чтобы обеспечить соответствие разработанной информационной системы целям и задачам организации.

В результате проведенного анализа методов преобразования требований в формальные модели было выявлено, что трансформации, как от требований в промежуточную модель, так и из промежуточной модели в формальную модель, описаны лишь на концептуальном уровне, что не дает возможность проверить их в действии [1, 2]. Соответственно встает вопрос о детализации, структуризации и уточнении правил трансформации. В рамках данной работы был разработан алгоритм преобразования требований в формальную модель, концептуально основанных на методе [3, 4].

Рассмотрим алгоритм преобразования требований в формальную модель.

На первом шаге преобразований требований необходимо применить правила для трансформации предложений на естественном языке в табличное представление, которое состоит из пяти колонок: словосочетание в начале предложения, подлежащее, сказуемое, дополнение и словосочетание в конце предложения. На втором шаге преобразований необходимо заполнить все пустые ячейки в полученном табличном представлении текста. Эти пустые ячейки образовались из-за того, что в требованиях содержатся предложения в страдательном залоге. Пассивной формой (страдательным залогом) называется такая конструкция предложения, при которой подлежащее не является действующим лицом (или предметом), а само подвергается действию со стороны дополнения (при этом дополнение может лишь подразумеваться, не будучи выражено в предложении). Для этих случаев необходимо преобразовать форму глагола из страдательного залога в действительный, и осуществить некоторые перестановки между колонками «Подлежащее» и «Дополнение» в табличном представлении, если это требуется. Как только заполнены все ячейки в табличном представлении, необходимо определить рабочие точки в системе – актеров. Актеры являются абстрактными компонентами системы, и они соответствуют подлежащему в предложениях. Необходимо рассматривать подлежащее лишь в тех предложениях, которые имеют глаголы, отличные от «быть». Алгоритм, представленный в третьем шаге, применяется к табличному представлению, в результате чего создается лист актеров. На следующем шаге необходимо вписать действия, выполняемые каждым из актеров, в их рабочие точки в том порядке, в котором они появились в тексте. Для этого необходимо для каждого актера последовательно пройти колонку «сказуемое» в табличном представлении и выписать действия, выполняемые одним «подлежащим». Наречия времени может изменить порядок действий, но сейчас они не являются объектом нашего исследования. На пятом шаге, используя таблицу элементов, необходимо спроектировать граф элементов и установить пути между последовательно выполняемыми действиями. Связи в данном графе устанавливаются с помощью элемента «соединительный путь». Следует отметить, что данный элемент отличается от термина «сообщение», используемого в UML-диаграмме последовательности. «Сообщение» в UML обозначает то, что один актер инициализирует действия другого. Инициализация выражается через глаголы, но для того чтобы отличать, какие глаголы являются сообщениями, а какие являются действиями, нам необходимы знания и интуиция. Автоматический анализ нуждается в интуиции человека эксперта, и для того, чтобы сделать работу проще, во внимание принимается следующая логика: все действия выполняются в рабочих точках. Путем соединения действий фактически представляется результат действий между рабочими точками [5].

Для апробации алгоритма преобразования требований в формальную модель будет использоваться пример требований, которые написаны для веб-ориентированной информационной системы интернет-магазина. Используя алгоритмы преобразования требований, осуществим преобразования небольшой части текстовых требований, которые были составлены для интернет-магазина в гибридную диаграмму деятельности.

Исходный текст требований имеет следующий вид: Покупатель заходит в интернет-магазин, чтобы сделать очередную покупку. Покупатель выбирает нужный товар. Товар добавляется в корзину и проверяется его наличие на складе. Если товар есть на складе, то оплата может происходить двумя способами: наложенным платежом и с помощью банковской карточки. Если оплата будет наложенным платежом, тогда покупатель оплачивает товар при его получении. Если оплата будет через банковскую карточку, то у покупателя запрашивается номер карты, срок действия, держателя карты. Покупатель вводит необходимые данные, чтобы инициировать передачу денежных средств со своего банковского счета. Если покупатель имеет недостаточно средств на счете для оплаты товара, тогда банк отклоняет запрос, и покупатель информируется об этом; иначе, если банк принял платеж, то генерируется электронный чек и посылается покупателю. Если товара нет на складе, покупателю предлагается выбрать другой товар.

Используя правила, описанные в первом шаге преобразований, необходимо записать предложенные требования в виде табличного представления (табл. 1)

Таблица 1

Шаг 1. Пример табличного представление текста

№	Словосочетание (в начале)	Подлежащее	Сказуемое	Дополнение	Словосочетание (в конце)
1		Покупатель	заходит	в интернет-магазин	, чтобы
			сделать	очередную покупку	.
2		Покупатель	выбирает	нужный товар	.
3		Товар	добавляется	в корзину	и
			проверяется	его наличие на складе	.

На втором шаге необходимо заполнить пустые ячейки данного табличного представления (табл. 2) в соответствии с правилами, описанными во втором шаге.

Таблица 2

Шаг 2. Трансформация страдательного залога в действительный

№	Словосочетание (в начале)	Подлежащее	Сказуемое	Дополнение	Словосочетание (в конце)
1		Покупатель	заходит	в интернет-магазин	, чтобы
2		ПОКУПАТЕЛЬ	делает	очередную покупку	.
3		Покупатель	выбирает	нужный товар	.
4		SYSTEM	добавляет	товар в корзину	и
5		SYSTEM	проверяет	его наличие на складе	.

Далее, в соответствии с третьим шагом, необходимо определить лист актеров, т.е. действующих лиц. Как видно из табл. 2, он будет следующим:

ЛА = (Покупатель, SYSTEM, Банк).

Теперь необходимо записать действия для актеров:

Л = [Покупатель (заходит-1, делает-2, выбирает-3, может оплачивает-9, вводит-12, инициирует-13, имеет_недостаточно_средств-14); SYSTEM (добавляет-4, проверяет-5, есть_на_складе-6, может происходит-7, будет_наложенным_платежом-8, будет_карточкой-10, запрашивает-11, информирует-16, генерирует-18, посылает_чек-19, предлагает_выбрать-21); Банк (отклоняет-15, принимает-17)].

На пятом шаге, используя таблицу элементов графа 1, необходимо произвести построение семантической сети (графа). Семантическая сеть представлена на рис. 1.

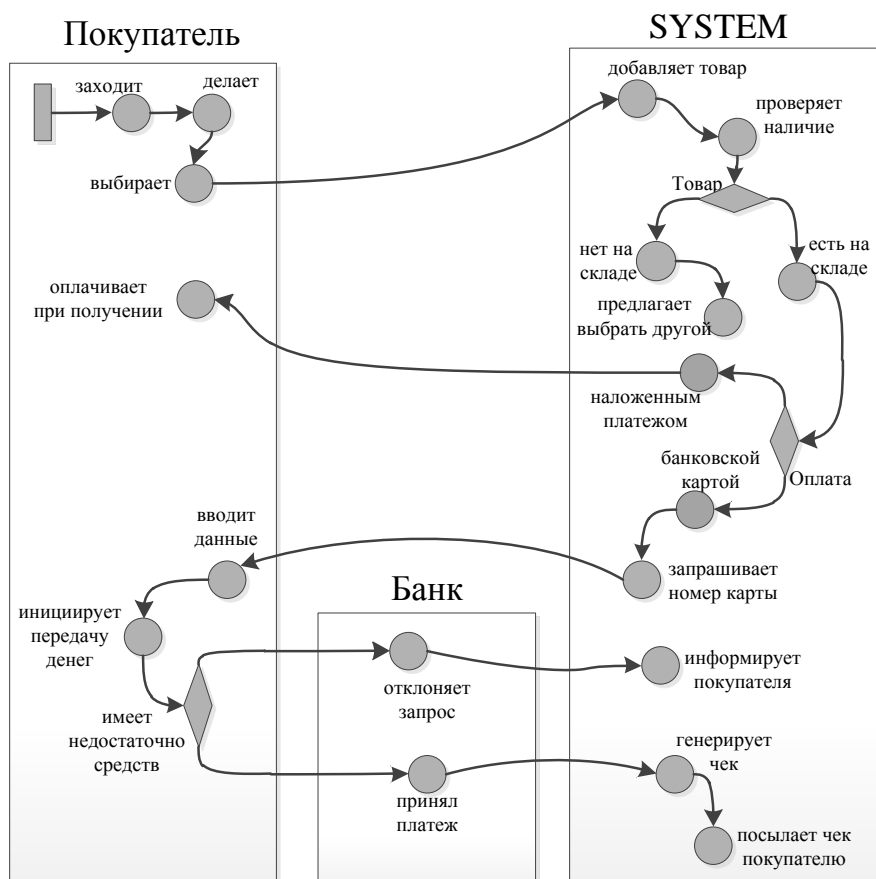


Рис. 1. Семантическая сеть

На последнем шаге метода, путем использования простых правил преобразования, получаем гибридную диаграмму деятельности (рис. 2).

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 знаний является очень важным, поскольку по-

зволяет визуально, быстро и точно определяется цель знания [6]. Также полученный граф может быть преобразован путем несложных изменений в гибридную диаграмму деятельности, из которой можно получить другие виды UML-моделей. Алгоритм преобразования текстовых требований в гибридную диаграмму деятельности может быть автоматизирован, поскольку написан в виде псевдокода [7, 8].

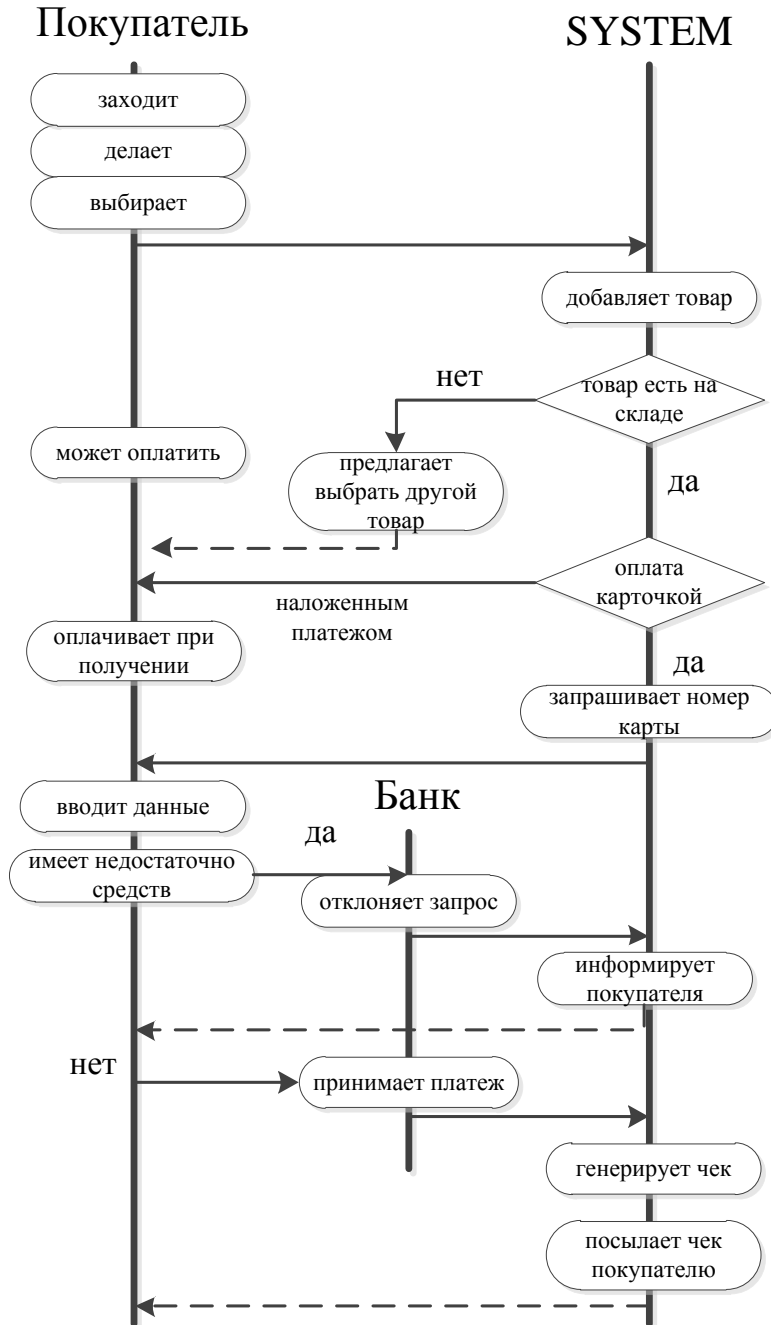


Рис. 2. Гибридная диаграмма деятельности

Разработанный алгоритм позволяет: извлекать основные и специфические предметные знания из текстов; получаемая формальная модель является полной, поскольку одновременно описывает как структуру и поведение системы, поскольку гибридная диаграмма деятельности также включает понятия актеров, бизнес-правил и сообщений; есть возможность преобразовать созданную формальную модель в ряд других моделей UML; табличное представление текста можно рассматривать как базу знаний, что может найти применение в проектах, использующих извлеченные знания из текста; графическое представление позволяет использовать принцип модульности при дальнейшем проектировании информационной системы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Рогозов Ю.И., Бутенков С.А., Свиридов А.С., Горбань Н.С.* Систематизация моделей жизненного цикла информационных систем в рамках схемы J. Zachman // Известия ЮФУ. Технические науки. – 2008. – № 1 (78). – С. 68-72.
2. *Рогозов Ю.И., Свиридов А.С., Горбань Н.С., Дубровский А.А., Жибулис Ю.А., Почечуев Н.В., Микита Р.М., Друшков С.А., Шевченко О.В.* Системный подход к построению информационных систем на основе жизненного цикла // Известия ЮФУ. Технические науки. – 2009. – № 1 (90). – С. 82-90.
3. *Abbott R.* Program design by informal English descriptions // Communications of the ACM. – 2006. – Vol. 26/11. – P. 882-894.
4. *Tao Yue, Lionel C. Briand, Yvan Labiche.* A systematic review of transformation approaches between user requirements and analysis models. Requirements Engineering – RE, 2011. – P. 75-99.
5. *Рогозов Ю.И., Свиридов А.С.* Концепция применения метадизайна и метамоделирования для разработки ИС // Известия ЮФУ. Технические науки. – 2012. – № 1 (126). – С. 147-152.
6. *Рогозов Ю.И., Свиридов А.С.* Метод построения интегральной модели жизненного цикла информационной системы // Телекоммуникации. – 2011. – № 3. – С. 8-17.
7. *Рогозов Ю.И., Бутенков С.А., Кодачигов В.И., Свиридов А.С., Микита Р.М.* Информационные ER++ модели – новый подход к интеграции основных этапов проектирования информационных систем // Известия ЮФУ. Технические науки. – 2006. – № 9-1 (64). – С. 70-74.
8. *Рогозов Ю.И., Свиридов А.С., Липко Ю.Ю.* Использование гиперметода разработки методов создания прототипов на примере построения структурно-независимых баз данных // Известия ЮФУ. Технические науки. – 2011. – № 1 (114). – С. 130-135.

Статью рекомендовал к опубликованию д.т.н., профессор В.Н. Иванченко.

Липко Юлия Юрьевна – Южный федеральный университет; e-mail: jullipko@gmail.com; 347900, г. Таганрог, ул. Чехова, 49, кв. 21; кафедра системного анализа и телекоммуникаций; доцент.

Lipko Yulia Yur'evna – Southern Federal University; e-mail: jullipko@gmail.com; 49, 21, Chekhov street, Taganrog, 347900, Russia; the department of system analysis and telecommunications; associate professor.