

А. Б. КУНГУРЦЕВ, канд. техн. наук, проф., ОНПУ, Одесса;

С. А. КАЛИНИНА, студентка, ОНПУ, Одесса;

Н. А. НОВИКОВА, ст. препод., ОНМУ, Одесса

МОДЕЛЬ ПРОЦЕССА ОПРЕДЕЛЕНИЯ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ

В статье автор ставит задачу имитации процесса специфицирования требований к программному продукту. В качестве решения предлагается тренажер определения требований; статья содержит краткое описание модели тренажера. Разработанная модель дает возможность обучаемому самостоятельно выбирать способ взаимодействия с сотрудниками организации-заказчика программного обеспечения и определять последовательность действий для формирования спецификации требований. Предлагается оценивать результат обучаемого путем анализа полученной спецификации требований и истории работы обучаемого.

Ключевые слова: требования к программному обеспечению; модель определения требований; способ выявления требований.

Введение. Успех программного продукта (ПП) во многом определяется правильной организацией работ по выявлению требований. Члены команды разработчиков ПП могут самостоятельно определять требования, либо привлекать специалиста-аналитика. В любом случае указанная деятельность требует не только профессиональной программистской подготовки, но и определенного жизненного опыта, умения общаться с людьми, аналитических способностей [1]. Все эти знания и умения специалисты-аналитики получают в процессе решения конкретных задач в разных предметных областях.

Для обучения процессу работы с требованиями в вузах предусмотрены различные традиционные виды занятий. Проблема заключается в том, что они позволяют студенту овладеть необходимым набором теоретических знаний, но не дают возможности промоделировать его взаимодействие с сотрудниками организации-заказчика программного продукта. Организация практических занятий в реальном процессе выявления требований практически неосуществима, в виду нерегулярности этого процесса, невозможности проведения «массовых» мероприятий, повторения определенного сценария и т. д.

Использование деловых игр [2,3], где роли заказчика и разработчика выполняют обучаемые, также некорректно, поскольку в основе выявления требований лежит взаимодействие двух (или более) лиц, с разной областью интересов и знаний. В работе [4] предложена методика обучения основам проектирования ПП с использованием ролевых компьютерных игр, однако не рассмотрены возможные реализации этапов этого процесса.

Таким образом, существует проблема создания модели реальной среды в рамках которой, обучаемый может «взаимодействовать» с заказчиком, выявлять заинтересованных лиц, проблемы, необходимые функции и другую информацию, необходимую для создания спецификации требований к программному продукту.

Цель работы. Целью работы является построение модели процесса сбора требований к программному обеспечению. Созданная модель ляжет в основу программной системы-тренажера для симулирования сбора требований.

Определим основные требования к модели.

1. Определение названия и краткой характеристики проекта, что даст возможность правильно выбрать направление дальнейшей деятельности обучаемого.

2. Представление структуры организации-заказчика в объеме, достаточном для выявления и получения доступа ко всем заинтересованным в проекте лицам.

3. Представление способов выявления требований, ассоциированных с заинтересованными лицами.

4. Представление документов и видов деятельности, ассоциированных с рабочими местами.

5. Предоставление возможности обучаемому «перемещаться» по организации, выбирать способы выявления требований, формировать фрагменты спецификации требований.

6. Предоставление обучаемому подсказок по методике и последовательности выявления требований, не связанных с конкретной организацией-заказчиком.

7. Хранение «истории» работы обучаемого с моделью.

8. Оценка результатов работы обучаемого путем анализа его действий и сравнения созданной им спецификации требований с эталоном.

Структура модели. В дальнейшем будем называть модель процесса определения требований главной моделью. Предложено представить главную модель в виде совокупности трех моделей:

$$M = \langle M_o, M_{inf}, M_p \rangle, \quad (1)$$

где M_o - структурная модель исследуемой организации (модель предметной области),

M_{inf} - информационная модель,

M_p - модель обучаемого.

Модель организации. Модель организации предназначена для представления структуры организации, для которой разрабатывается информационная система. Модель имитирует рабочие места, подразделения, связи подчинения и взаимодействия, существующие в организации-заказчике. Представим M_o в виде множества узлов:

$$M_o = \{U_i\} i = 1, n \quad (2)$$

Каждый узел представляет собой некоторую структурную единицу организации:

$$U_i = \langle N, D, Lh, Ls, Lc \rangle \quad (3)$$

где N - название (идентификатор) узла,

D - структурное подразделение, в которое входит узел,

Lh - множество связей «подчиненный – начальник»,

Ls - множество связей «начальник – подчиненный»,

Lc - множество связей сотрудничества.

Каждая связь представлена кортежем

$$\langle N, St \rangle, \quad (4)$$

где N - узел, с которым связан данный узел,

St - состояние связи (доступна для перехода - true, закрыта – false).

Информационная модель. Информационная модель служит для представления информации, которую обучаемый может получить на каждом

рабочем месте, алгоритмов управления процессом обучения, эталонов формируемых требований.

$$M \text{ inf} = \langle R, Pm, Doc \rangle, \quad (5)$$

где $R = \{R_i\}_{i=1, n}$ - множество вариантов получения информации о требованиях, связанных с каждым узлом модели организации,

Pm - система подсказок,

Doc - определение документации, которую должен сформировать обучаемый.

Предлагается представить R_i в виде кортежа

$$R_i = \langle N_i, fw, V \text{ inf} \rangle, \quad (6)$$

где N_i - текущий узел,

fw - функция демонстрации возможных действий обучаемого в текущем узле,

$V_{\text{inf}} = \{V_j\}, \quad j = 1, n\}$ - множество вариантов получения информации обучаемым и управление его дальнейшим поведением.

Предлагается каждый элемент V_j представить в виде

$$V_j = \langle Ver, Val, fa, fn \rangle, \quad (7)$$

где Ver - одна из множества версий получения требований ($Ver \in MVer$),

Val - эталонный выбор,

fa - функция анализа выбора обучаемого и формирования элемента требований в редакции обучаемого,

fn - функция навигации, управляющая связями между узлами в зависимости от результатов деятельности обучаемого.

В $M \text{ inf}$ используются следующие версии получения требований

$$MVer = \{Vi, Vuc, Vd, Va\}, \quad (8)$$

где Vi - взятие интервью;

Vuc – фиксирование основных вариантов использования;

Vd – определение документов, участвующих в документообороте;

Va - прототипирование.

Документация представлена двумя составляющими

$$Doc = \langle Cont, Seq \rangle$$

$Cont$ представляет собой содержание спецификации требований. Примерная структура спецификации требования выглядит следующим образом:

1. Цель создания системы
2. Затраты на разработку и сроки создания продукта
3. Позиционирование
 - 3.1. Характеристика бизнес-процесса в настоящее время
 - 3.2. Определение проблемы и подпроблем
4. Описание пользователей
 - 4.1. Состав пользователей
 - 4.2. Профили пользователей
 - 4.3. Ключевые потребности пользователей
5. Функциональные требования
 - 5.1. Возможности продукта
 - 5.2. Ключевые прецеденты
6. Нефункциональные требования к программному продукту

- 6.1. Системные требования
- 6.2. Эксплуатационные требования
- 6.3. Требования надёжности
- 7. Требования к документации
- 7.1. Руководства пользователя
- 7.2. Интерактивная справка
- 7.3. Руководства по установке и конфигурированию
- 8. Глоссарий

Seq содержит допустимую последовательность заполнения разделов спецификации требований.

Модель обучаемого

Модель обучаемого служит для

- идентификации обучаемого,
- формирования требований о системе по версии обучаемого,
- восстановления позиции обучаемого после перерыва в работе с моделью,
- оценки работы обучаемого с моделью, накопления статистических данных о процессе обучения.

Представим модель обучаемого в виде кортежа

$$M_p = \langle N_s, H_i, R_s \rangle, \quad (9)$$

где N_s - фамилия (идентификатор) обучаемого,

H_i - история работы обучаемого с моделью,

R_s - результаты работы обучаемого с моделью (спецификация требований).

История работы обучаемого представляется в виде

$$H_i = \langle N_i, Val, Seq_i \rangle, \quad (10)$$

где N_i - множество узлов, к которым получил доступ обучаемый,

Val - оценки его деятельности,

Seq_i - разделы эталона документации, доступные для обучаемого.

В начале работы обучаемого с системой $Cont$ копируется в спецификацию требований обучаемого R_s . В процессе работы выявленные обучаемым требования заносятся в соответствующие разделы R_s .

Пример вариантов получения информации для требований. Пусть модель организации соответствует университету, для которого предполагается создание автоматизированной системы «Абитуриент». Предполагается, что ранее процессы приёма документов у абитуриентов, организации экзаменов, определения проходных баллов и распределения новых студентов по факультетам, специальностям и группам не были автоматизированы, однако, электронные документы использовались. Требуется разработать ИС «Абитуриент» для автоматизации указанных процессов.

Пусть в вершине иерархии узлов модели организации (3) находится проректор по учебно-методической работе (ПУМ). С этим узлом связана контекстно-зависимая подсказка P_m , где указано название проекта, ответственное лицо за разработку проекта со стороны заказчика – ПУМ, и содержатся общие рекомендации о начале процесса выявления требований в соответствии с эталоном документации.

Функция демонстрации возможных действий обучаемого f_w представит следующие возможности обучаемому:

1. Проведение интервью.

2. Выбор одного или нескольких вариантов использования (прецедентов) из предлагаемого списка.

Рассмотрим принцип работы обучаемого с интервью [5]. Обучаемый должен выбрать вопросы из предлагаемого списка. Вопросы подразделяются на правильные и неправильные. В зависимости от данных в истории работы обучаемого с моделью H_i , ему оказываются доступны определенные разделы спецификации требований. В соответствии с этим, правильные вопросы подразделяются на своевременные и несвоевременные. В зависимости от назначения вопросы могут служить для выявления требований, либо для обеспечения навигации обучаемого по модели организации. Если обучаемый выбирает вопрос из категории выявления требований, то следует также указать раздел эталона требований, куда следует поместить ответ.

Обучаемый может увидеть ответ на вопрос только после выбора этого вопроса и указания раздела документа (в зависимости от категории вопроса). В табл.1 приведен фрагмент интервью, связанного с узлом ПУМ. Оценки выбора вопросов выставлены исходя из того, что в нашем случае обучаемый начинает работать в системе.

Таблица 1 - Фрагмент интервью для узла ПУМ

| Вопрос | Раздел документации (определяется обучаемым) | Оценка вопроса (определяется системой) | Ответ |
|---|--|--|---|
| Для чего Вы заключаете контракт на разработку системы «Абитуриент»? | 1 (правильно) | правильно | У нас существуют проблемы с получением полной, достоверной и своевременной информации об абитуриентах |
| Какие функции на Вашем рабочем месте должна выполнять система «Абитуриент»? | 1(неправильно) 5.1(правильно) | Несвоевременный | Получение данных о поданных заявлениях, о прохождении экзаменов, о планах набора, о категориях абитуриентов |
| Какие средства Ваша организация может выделить на разработку новых ПП? | Отсутствует | неправильный | Ответ отсутствует |
| Какие средства вуз планирует выделить на разработку системы «Абитуриент»? | 2(правильно) | правильный | Планируемые расходы составляют ... |
| Какова организационная структура Вашего университета? | Отсутствует | неправильный | Ответ отсутствует |
| Какие подразделения участвуют в работе с абитуриентами? | 4.1(правильно) | правильно | Это приёмная комиссия, учебно-методический отдел, ОК студентов, деканаты |
| Можете ли Вы меня представить начальникам подразделений, заинтересованным в системе «Абитуриент»? | Навигация | правильно | Безусловно, сейчас я свяжусь с ними по телефону |

Рассмотрим принцип работы обучаемого с прецедентом [6]. Описание прецедента рассматривается как отдельный документ, на который должна быть сделана ссылка из определенного раздела документации. Раздел документации позволяет оценить своевременность использования прецедента. Обучаемому представлен заголовок описания прецедента, который содержит достаточно информации для принятия решения о своевременности и месте включения прецедента в спецификацию требований. Ниже приведен пример заголовка. Курсив определяет структуру прецедента. Комментарии заключены в скобки.

1. *Название прецедента*: Получение данных о поданных заявлениях
2. *Основное действующее лицо*: проректор по учебно-методической работе
3. *Уровень*: пользователя (уровень пользователя или уровень системы)
4. *Участники и интересы*: проректор желает получить оперативные данные о количестве поданных заявлений на указанную дату, специальность, факультет, форму обучения с целью возможного влияния на процесс подачи заявлений.
5. *Предусловие*: проректор идентифицирован.
6. *Минимальные гарантии*: получение интегрированной информации (что получает система при выполнении любого сценария)
7. *Гарантия успеха*: получение информации в соответствии с комбинацией значений параметров (что получает система при выполнении успешного сценария)
8. *Триггер*: проректор выбрал режим «Приёмная комиссия» - «Поданные заявления» (событие, определяющее возникновение прецедента)

Выводы. Разработана математическая модель, позволяющая для обучаемого организовать процесс выявления требований к программному продукту, который в значительной степени приближается к реальному. Разработаны структуры документов, соответствующие общепринятым требованиям, а также процедуры их использования и формирования, что позволяет формировать содержательные отчёты по результатам обучения, контролировать процесс обучения, фиксировать ошибки и оценивать знания и умения обучаемого. Предлагаемая модель может быть использована для различных предметных областей.

Список литературы: 1. *Леффингуэлл Д.* Принципы работы с требованиями к программному обеспечению. / *Дин Леффингуэлл, Дон Уидриг.* - М., СПб. : Вильямс, 2002. - 450с. 2. *Бельчиков Я.* Деловые игры. / *Я. М. Бельчиков, М. М. Бириштейн.* - Рига : АВОТС, 1989 – 304с. 3. *Веденина В.* Деловая игра и ее возможности [Электронный ресурс]. – Режим доступа: URL: <http://tolerance.ru/teacher/kabinet/business-game.html> (дата обращения: 26.02.2013). – Название с экрана. 4. *Кунгурцев А.* Методика проведения обучения основам проектирования программных систем в виде ролевых компьютерных игр. Годишник на технически университет.[в 3т.]/ *Кунгурцев А., Блажко А., Марулин С.* – Варна : Варна, 2010 - . – Том 1: Сборник доклади от юбилейна научна конференция с международно участие. – 2010. - 123-126с 5. *Лукина М.* Технология интервью. Учебное пособие для вузов./*Лукина М.* - М.: Аспект Пресс, 2003 – 480с 6. *Коберн А.* Современные методы описания функциональных требований к системам./ Алистер Коберн – М.:Лори, 2011 - 263с.

Поступила в редколлегию 01.06.2013

УДК 004.855.5

Модель процесса определения требований к программному продукту/ А. Б. Кунгурцев, С. А. Калинина, Н. А. Новикова // Вісник НТУ «ХПІ». Серія: Нові рішення в сучасних технологіях. – Х: НТУ «ХПІ», – 2013. - № 38 (1011). – С.55-61 . – Бібліогр.: 6 назв.

У статті автор ставить задачу імітації процесу специфікування вимог до програмного продукту. В якості рішення пропонується тренажер визначення вимог; стаття містить короткий опис моделі тренажера. Розроблена модель дає можливість тому, хто працює з тренажером, самостійно вибирати спосіб взаємодії з співробітниками організації-замовника програмного забезпечення і визначати послідовність дій для формування специфікації вимог. Пропонується оцінювати результат студента шляхом аналізу отриманої специфікації вимог і історії роботи студента.

Ключові слова: вимоги до програмного забезпечення; модель визначення вимог; спосіб виявлення вимог.

In the article, the author poses the problem of simulate the process of specifying requirements for software. The requirements definition simulator is proposed as a solution; the article contains a brief description of the simulator model. The developed simulator model enables students to select the method of interaction with employees of the organization-client of software and to determine steps for the formation of the requirements specification. Proposed to estimate the student's result by analyzing the final requirements specification and the student's work history.

Keywords: software requirements, requirements definition model, a method for detecting requirements.

УДК 004.77

Є. С. САКАЛО, канд. техн. наук, доц., ХНУРЕ, Харків

АНАЛІЗ ПРОЦЕСІВ РОБОТИ ДИНАМІЧНИХ ОБ'ЄКТІВ

Розглянуті існуючі алгоритми управління динамічним об'єктом, об'єднання принципів їх роботи, виявлення їх недоліків та розробка власного алгоритму, котрий частково покриє данні недоліки

Ключові слова: нейронна мережа, нейроконтроллер, перцептрон, нейроемулятор

Вступ. Традиційні методи, що використовувалися для управління динамічними об'єктами, такі як математичне моделювання та модель авторегресії, модель кристалічної сітки, показали позитивні якості, як аналіз явища, що відбувається в заданій полосі частот, але вони часто не дають достатню точність, що необхідна для управління об'єктом. Також вони основані на моделях «чорного ящика», коли змінні та параметри не мають жодного фізичного змісту. Нейроуправління динамічними об'єктами є новим перспективним напрямом, що знаходиться на стику таких різних дисциплін, як автоматичне управління, штучний інтелект, нейрофізіологія. Нейронні мережі мають ряд унікальних властивостей, які роблять їх потужним інструментом для створення систем управління: здатністю до навчання на прикладах і узагальнення даних, адаптуватися до змін властивостей об'єкта управління та зовнішнього середовища, придатністю для синтезу нелінійних регуляторів, високої стійкістю до пошкоджень своїх елементів завдяки паралелізму нейромережової архітектури. Нейронні мережі дозволяють реалізувати будь-який необхідний для процесу нелінійний алгоритм управління при неповному, неточному описі об'єкта управління[1-2].

Системи управління повинні бути простими у використанні та для розуміння, та мати такі властивості, як навчання, гнучкість, нелінійність та стійкість. Саме такі умови являються основними для розробки програмної системи, що дозволить управляти динамічними об'єктом у реальному часі. Нейронні мережі показали свої властивості під час експериментів у розпізнаванні образів, так як мають можливість навчатися у відношенні «вхід-вихід», що дозволяє представити більш прості

© Є. С. САКАЛО, 2013