

УДК 519.2

В. А. ШЕХОВЦОВ, канд. техн. наук, доцент НТУ «ХПІ»

РЕЖИМИ ВИКОРИСТАННЯ ІНТЕРАКТИВНОГО СЕРЕДОВИЩА ПІДТРИМКИ ІНТЕРАКТИВНОГО ЗБОРУ ВИМОГ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Збір вимог до нових програмних проєктів традиційно спирається на інтерв'ю з зацікавленими особами. Проведення такого збору є складною задачею, оскільки очікувані користувачі досить часто не можуть пояснити, які системні якості їм треба отримати. Ми пропонуємо обійти цю проблему шляхом конструювання та використання спеціальної програмної системи – інтерактивного імітаційного середовища (ІС). Користуючись ІС, потенційні користувачі можуть наглядно досліджувати можливі засоби організації комп'ютерної підтримки їх праці. У роботі розглянуто основні режими використання ІС.

Сбор требований к новым программным проектам традиционно опирается на интервью с заинтересованными лицами. Проведение такого сбора является сложной задачей, поскольку ожидаемые пользователи довольно часто не могут объяснить, какие системные качества им нужно получить. Мы предлагаем обойти эту проблему путем конструирования и использования специальной программной системы – интерактивной имитационной среды (ИИС). Пользуясь ИИС, потенциальные пользователи могут наглядно исследовать возможные средства организации компьютерной поддержки их работы. В работе рассмотрены основные режимы использования ИИС.

Requirements elicitation for new computer applications rests traditionally on interviewing stakeholders. That makes getting the user requirements right more complicated than is necessary because the anticipated users not necessarily are good at talking about the kind of computer aid and its qualities they would appreciate. We suggest bypassing that problem by constructing and using POSE a parameterized online simulation environment within which anticipated users can explore ways of computer support for their work and which then makes it easier to talk about a potential computerized aid and the qualities thereof. The paper describes main POSE working modes.

1. Вступ. У цій роботі ми розглядаємо підхід до збору вимог якості до програмних систем. Такий збір у випадку нових програмних проєктів традиційно спирається на інтерв'ю з зацікавленими особами (ЗО). Його проведення є складним завданням, оскільки ЗО досить часто не можуть пояснити, які системні якості їм треба отримати. Ми пропонуємо розробити й використовувати для збору вимог якості спеціальне програмне середовище – *ІС (інтерактивне імітаційне середовище)*, у якому ЗО можуть взаємодіяти з імітаційною моделлю системи, що розроблюється (СщР) в рамках процесів її використання, теж заданих імітаційними моделями. У ході цієї роботи вони можуть оцінювати якісні характеристики, задані моделлю, визначаючи тим самим (або уточнюючи для себе) якісні вимоги до вихідної системи. Це дає ЗО можливість більш вільно оцінювати якість системи та потенційну підтримку, яку вони очікують отримати. Ми очікуємо, що використання ІС спростить формулювання вимог якості, природно, що для підтвердження цього будуть потрібні додаткові емпіричні дослідження. Ця робота присвячена

розв'язанню завдання визначення основних режимів роботи імітаційного середовища.

План роботи. У розділі 2 ми оцінимо сучасні методи збору вимог якості. У розділі 3 буде наведена постановка завдання визначення основних режимів розробки ПС, розділ 4 буде присвячено опису цих режимів, у розділі 5 будуть розглянуті типи користувачів системи. Наприкінці роботи будуть наведені висновки та план подальших досліджень.

2. Існуючі технології збору вимог якості. У цьому розділі ми розглянемо існуючі технології збору вимог якості.

Технології, у яких задіяні зацікавлені особи. Ці технології використовують традиційні підходи (інтерв'ю, методи "мозкового штурму", списки для перевірки й т.ін.). Звичайно для цього набір вимог якості структурують, після чого це подання використовується як підказка для зацікавлених осіб. Технології, орієнтовані на мети [1] класифікують вимоги якості відповідно до мет системи. Словникові технології [2, 3] використовують для цього спеціальні словники. Підхід Quality Attribute Workshops [4] визначає організаційне середовище, ціль якого – ідентифікація атрибутів якості даної системи (характеристик, які задовольняють вимогам якості) шляхом розробки їхніх інтерпретацій для кожного конкретного випадку (сценаріїв), ідентифікації сценаріїв, визначення їхніх характеристик і завдання пріоритетів.

Технології видобутку вимог. Вихідними даними для цих технологій є специфікації вимог, задані у вигляді тексту природною мовою або в структурованому виді. Для автоматизованого одержання вимог із цих документів передбачається використовувати засоби обробки природної мови, засобу видобутку інформації (Information Retrieval) і інші подібні технології. У методі Theme/Doc [5] виробляється обробка специфікацій вимог шляхом пошуку ключових слів, які використовуються для генерації діаграм для етапу проектування. У роботі [6] використовуються технології видобутку інформації (зокрема, робота з регулярними вираженнями) для пошуку вимог, які ставляться до конкретної якості системи. Дотримуючись підходу [7], визначається набір слів-індикаторів (ключових слів), цей набір навчається на реченнях-прикладках (визначаючи ваги, що задають відповідність термінів категоріям вимог), після чого пропозиції документа класифікуються по категоріях вимог на підставі інтегральної ваги всіх їхніх індикаторів.

Підходи для конкретної предметної області. Існують підходи, які орієнтовані на збір певних класів вимог якості. Так, наприклад, проект TROPOS [8] підтримує деякі технології й програмні засоби, призначені для збору вимог безпеки, такі як діаграми безпеки, обмеження безпеки й т.д. Варіанти невірнього використання (misuse cases [9]) допомагають у зборі вимог безпеки, описуючи сценарії некоректного або несанкціонованого використання системи. Інші підходи орієнтовані на збір вимог до продуктивності [10], до надійності й т.ін.

3. Постановка завдання визначення основних режимів роботи ІС. У цьому розділі ми зупинимось на недоліках наведених підходів, покажемо, яким чином дослідження цих недоліків може привести до ідеї ІС та сформулюємо постановку завдання визначення основних режимів його використання.

Наведені технології збору вимог якості не вільні від недоліків. Так, основний недолік підходів, у яких задіяні ЗО, полягає в тому, що багато ЗО не можуть обговорювати вимоги якості без попереднього досвіду практичної роботи із системою. Недоліком технологій видобутку вимог є обмежена участь зацікавлених осіб у процесі одержання вимог. На практиці, після того, як процес збору завершений, вимоги все одно повинні пройти додаткову верифікацію зацікавленими особами. Використання підходів для конкретної предметної області, як правило, обмежене цією областю,

Для розв'язання цих проблем необхідно розробити інтерактивне імітаційне середовище, яке б дозволяло ЗО придбати досвід взаємодії із СщР якомога раніше. Для розробки такого середовища потрібно перш за все визначитися з режимами його роботи. Ці режими повинні забезпечувати досягнення основних цілей розробки ІС, архітектура ІС повинна бути зпроектована з урахуванням необхідності їхньої підтримки.

4. Режими використання ІС. Ми стверджуємо, що ІС може використовуватися в чотирьох режимах:

1. *режимі конструювання моделі*, коли відбувається адаптація середовища імітаційного моделювання для конкретного завдання;
2. *режимі виконання моделі*, призначеному для інтерактивного збору оцінок якості СщР у ході взаємодії ЗО з імітаційними моделями, для яких визначені різні значення ключових параметрів;
3. *режимі аналізу вимог*, що використовується для аналізу результатів експериментів (оцінок якості) і формулювання вимог, зібраних у ході цих експериментів;
4. *режимі перевірки доцільності*, коли відбувається перевірка, чи є доцільним розгортання ІС для конкретної організації, яка оперує у визначених ресурсних обмеженнях.

Базова ідея. Ми розглядаємо організацію як систему, у якій виконується певний набір бізнес-процесів. Як наслідок, для створення імітаційної моделі організації нам необхідно змодельовати її структуру, ресурси (у тому числі людські), які вона використовує, її бізнес-процеси.

Для моделювання процесів ми пропонуємо використовувати програмний засіб, розроблений у результаті досліджень в області асоціативного пошуку й повторного використання бізнес-процесів: асемблер процесів (АП) [11]. АП зберігає моделі процесів, описані на різних мовах моделювання процесів (ММП), і дає можливість відображати моделі з одної ММП на іншу за умови,

що виразні засоби мови дозволяють виконувати таке відображення, і в системі зареєстровані відповідного драйвера для відображення.

Зовнішні системи можуть одержувати доступ до АП через інтерфейс веб-сервіса. У цьому випадку як така зовнішня система виступає ІС. З погляду ІС, процес взаємодії з АП виглядає в такий спосіб. ІС генерує запит на одержання моделі процесу, у якому вказується використовувана у ньому ММП, а також інші параметри відповідно до метаданих АП. АП одержує найбільш підходящі моделі з використанням засобів асоціативного пошуку й, якщо отримана модель описана на ММП, з якою ІС не може працювати, відображає цю модель на потрібну мову, якщо для неї в АП є драйвер.

Якщо згадані моделі доступні, доступна інформація щодо навантаження, тобто інформація з розподілу ймовірностей активізації процесів, необхідне час обробки для цих процесів, інформація з утилізації ресурсів і т.д., у принципі можна побудувати імітаційну модель всієї організації. Основна мета розробки ІС - дати можливість потенційним користувачам СщР одержати досвід взаємодії з імітаційною моделлю, що опирається на ключові параметри використання СщР. Ми маємо намір забезпечити взаємодію серверних компонентів імітаційної моделі й моделей програмних компонентів, через які користувачі задають свої запити. Таким чином, ми можемо надати користувачам можливість одержати враження від випробування функціональності СщР, наближені до реального досвіду, з опорою на статистичний аналіз, закладений в імітаційній моделі.

Режим конструювання моделі. Однієї з найбільш важливих завдань, що повинні вирішити користувачі ІС, є початкове визначення області передбачуваного моделювання, тобто визначення того, що вважати моделюваною організацією. Це завдання вимагає розуміння бізнес- процесів, для участі в які передбачається використовувати СщР. У режимі конструювання моделі, отже, необхідно визначити й формалізувати бізнес-процеси даної організації. У даній роботі ми не будемо обговорювати, як це може бути зроблено, підходи до розв'язання цього завдання є об'єктом подальшого дослідження.

Користувачі ІС надають всі початкові дані, які необхідні для проведення імітаційних експериментів. Зокрема, у цьому режимі необхідно задати структуру організації, включаючи ролі персоналу, а також інформацію про навантаження. Даний процес називається параметризацією ІС, він охоплює типові дані про навантаження, такі як частотні характеристики, розподіли появи подій, доступність персоналу й інших ресурсів, ймовірність виникнення різних катастроф, ступінь серйозності цих катастроф і т.д.

Режим виконання моделі. У режимі виконання моделі виробляється імітаційне моделювання процесів, певних у режимі адаптації. На вибір користувачів, моделі компонентів СщР, зареєстрованих в ІС, можуть супроводжуватися графічною анімацією або виконуватися як реальні компоненти. Можливе перевизначення значень параметрів, заданих при параметризації ІС. Крім того, користувачі ІС можуть залишати коментарі із приводу їхніх

вражень від характеристик якості системи. Для цього пропонується дати їм можливість формально оцінювати зареєстровані компоненти СщР. По-перше, вони можуть дати оцінку версії компонента СщР по десятибальній шкалі. Ця оцінка повинна відповідати впевненості користувачів у вигоді використання даної версії. По-друге, користувачі можуть із кожною парою версій СщР (V1,V2) зіставити предикати "*набагато краще(V1,V2)*", "*краще(V1,V2)*", "*небагато краще(V1,V2)*", і "*еквівалентні(V1,V2)*".

Режим аналізу вимог. У цьому режимі користувачі працюють з даними, що отримані в режимі виконання моделі. Для таких користувачів (як правило, це – інженери зі збору вимог), ІС надає ряд звітів із заздалегідь проаналізованими експериментальними даними. Кожний такий звіт включає номер версії моделі, характеристики навантаження, структуру організації й т.ін. У цьому режимі також виконується перетворення оцінок ЗО у вимоги якості, для чого повинно бути сформульоване завдання багатокритеріального вибору.

Режим перевірки доцільності. Цей режим дозволяє аналітику досліджувати економічну доцільність розгортання запропонованого середовища та виконувати необхідні дії, якщо така доцільність не визначена, або якщо розгортання ІС є занадто важким. До цих дій відносяться коригування ресурсних обмежень або спрощення конфігурації ІС згідно з методологією системної оптимізації [12], а також розглядання можливості вибору альтернативної технології збору вимог якості, якщо таке коригування неможливе. При оцінюванні альтернативних технологій збору вимог якості будуть використовуватись критерії, що базуються на певній моделі якості, ці критерії будуть інтегровані у задачу багатокритеріального прийняття рішень.

5. Типи користувачів ІС. Експерти ІС роблять адаптацію ІС. Вони підрозділяються на експертів по бізнес- процесам (БЕ) і експертів по системі, що розроблюється (СЕ). БЕ задають процеси використання системи, вони є фахівцями в області бізнес-процесів фірми й можуть перетворювати свої знання в дії, вироблені в режимі з. Такі експерти звичайно не знайомі з усіма тонкощами розробки СщР і в більшості випадків розглядають її як "чорну шухляду". СЕ проєктують і створюють моделі СщР і прототипи компонентів СщР, після чого інтегрують їх в ІС. Ми припускаємо, що в ролі СЕ будуть виступати кваліфіковані розроблювачі.

ЗО – це користувач, що може оцінювати якісні характеристики СщР у процесі інтерактивної взаємодії з імітаційною моделлю. Деякі ЗО добре знайомі з основним бізнесом фірми й розуміють, як до цього бізнесу ставиться СщР. Ми очікуємо, що такі СщР мають знання об бізнес- процесах передбачуваного використання СщР і зовнішніх вимогах до якості СщР, таких, як продуктивність або доступність. Інші ЗО є фахівцями в області інформаційних технологій, які відповідають за організацію інформаційної інфраструктури фірми, тобто за виконання установки, відновлення або підтримки системи. Вони, швидше за все, нададуть вимоги, які ставляться до

якісних характеристик процесу розробки, що зачіпає СцР. Прикладами таких вимог є вимоги до захисту, супроводжуваності, здатності до розвитку.

Системний адміністратор (СА) робить регулярні дії по підтримці працездатності ІС. Він відповідає за адміністративні завдання, такі як резервне копіювання або щоденну підтримку. Такі адміністратори також відповідають за успішне виконання імітаційного моделювання й виконують ролі адміністраторів такого моделювання (виконують прогони моделей, інструктують і навчають ЗЛ, дозволяють конфлікти, роблять із і т.д.). Крім того, СА повинен могли задавати параметри імітаційної моделі.

6. Висновки й перспективи подальшої роботи. У ході виконання дослідження були запропоновані режими використання інтерактивного середовища збору вимог якості, які дозволяють йому надавати ЗО можливість ознайомитись з модельованими якостями СцР до того, як вони братимуть участь безпосередньо у зборі вимог.

Ми збираємося реалізувати перший прототип ІС. Після цього нашою метою є перевірка коректності цього прототипу. У ході цієї перевірки ми сподіваємося показати, що ІС (принаймні, для певних випадків) є ефективнішим засобом збору вимог якості в порівнянні з іншими методами.

Список літератури: 1. Non-functional requirements in software engineering / L. Chung, B. A. Nixon, E. Yu, J. Mylopoulos. – Boston : Kluwer, 1999. – 476 p. – ISBN 978-0-7923-8666-7. 2. Cysneiros L. M. Non-Functional Requirements Elicitation / L. M. Cysneiros, E. Yu // J. C. Sampaio do Prado Leite, J. H. Doorn (Eds.). Perspectives on Software Requirements. – Boston : Kluwer, 2004. – P. 115-138. – ISBN 978-1-4020-7625-1. 3. Шеховцов В. А. Обработка требований качества на ранних этапах разработки программного обеспечения / В. А. Шеховцов // Вестник НТУ «ХПИ». Сборник научных трудов. Тематический выпуск «Системный анализ, управление и информационные технологии». – Харьков : НТУ «ХПИ». – 2007. – № 41. – С. 162-170. 4. Quality Attribute Workshops (QAWs), Third Edition. Technical Report CMU/SEI-2003-TR-016 / M. R. Barbacci, R. Ellison, A. J. Lattanze [et al.]. – Pittsburgh, PA : Carnegie Mellon University, Software Engineering Institute. – 2003. – 38 p. 5. Clarke, S. Aspect-oriented analysis and design: the Theme approach / S. Clarke, E. Baniassad. – Reading, MA : Addison-Wesley, 2005. – 400 p. – ISBN 0-321-24674-8. 6. Rosenhainer, L. Identifying crosscutting concerns in requirements specifications [Electronic resource] / L. Rosenhainer // Workshop on early aspects. – 2004. – Vancouver, Canada. – <http://trese.cs.utwente.nl/workshops/oopsla-early-aspects-2004/Papers/Rosenhainer.pdf> 7. Cleland-Huang, J. Automated classification of non-functional requirements / J. Cleland-Huang, R. Settini, X. Zou, P. Solc // Requirements Engineering. – 2007. – Vol. 12, Issue 2. – P.103-120. 8. Bresciani, P. TROPOS: An Agent-Oriented Software Development Methodology / P. Bresciani, P. Giorgini, F. Giunchiglia [et al.] // Journal of Autonomous Agents and Multi-Agent Systems. – 2004. – Vol.8, Issue 3. – P. 203–236. 9. Alexander, I. Misuse Cases: Use Cases with Hostile Intent / I. Alexander // IEEE Software. – 2003. – Vol. 20, Issue 1. – P. 58-66. 10. Nixon, B. A. Managing Performance Requirements for Information Systems / B. A. Nixon // Proceedings of the 1st international workshop on Software and performance. – New York : ACM Press. – 1998. – P. 131-144. – ISBN 1-58113-060-0. 11. Zlatkin, S. Towards Amplifying Business Process Reuse / S. Zlatkin, R. Kaschek // J. Akoka, S. W. Liddle, I.-Y. Song [et al.] (Eds.). Perspectives in Conceptual Modeling: ER 2005 Workshops, Lecture Notes in Computer Science, Vol. 3770. – Berlin-Heidelberg : Springer, 2005. – P.364-374. – ISBN 3-540-29395-7. 12. В. М. Глушков. О системной оптимизации. / В. М. Глушков // Кибернетика. – 1980. – № 5. – С. 89-90.

Поступила в редколлегию 23.03.09