

УДК 004.89:004.9

DOI: 10.15587/1729-4061.2019.154074

Розроблення інтелектуального агента для аналізу нефункційних характеристик у специфікаціях вимог до програмного забезпечення

Т. О. Говорущенко, О. О. Павлова, М. А. Боднар

Сьогодні актуальною задачею є автоматизований аналіз специфікацій вимог до програмного забезпечення (ПЗ) на предмет достатності інформації щодо нефункційних характеристик-складових якості ПЗ. Проведений аналіз відомих інтелектуальних агентів на основі онтологічного підходу показав, що відомі агенти не розв'язують задачу кількісного оцінювання достатності інформації у специфікації вимог до ПЗ для визначення нефункційних характеристик ПЗ.

Задачею даного дослідження є реалізація інтелектуального агента на основі онтологічного підходу для аналізу інформації щодо нефункційних характеристик у специфікаціях вимог до ПЗ.

Розроблено модель діяльності інтелектуального агента на основі онтологічного підходу для оцінювання специфікацій вимог до ПЗ. Вона відображає особливості оцінювання достатності інформації для визначення нефункційних характеристик-складових якості ПЗ. Розроблена модель є теоретичним підґрунтям для реалізації інтелектуального агента на основі онтологічного підходу для оцінювання специфікацій вимог до ПЗ.

Реалізовано інтелектуальний агент на основі онтологічного підходу для оцінювання інформації щодо нефункційних характеристик у специфікаціях вимог до ПЗ. Реалізований агент формує висновок про достатність або недостатність інформації щодо нефункційних характеристик-складових якості ПЗ у специфікації вимог до реального ПЗ. Крім цього, він кількісно оцінює рівень достатності інформації у специфікації вимог до реального ПЗ для визначення кожної нефункційної характеристики ПЗ та для визначення всіх нефункційних характеристик-складових якості ПЗ разом. Агентом надається список атрибутів, якими варто доповнити специфікацію вимог для підвищення рівня достатності її інформації, а також візуалізація прогалів у знаннях про всі нефункційні характеристики-складові якості ПЗ.

Результати функціонування реалізованого агента в комплексі забезпечують підвищення рівня достатності інформації у специфікації вимог до ПЗ. Розроблений інтелектуальний агент дозволяє частково усунути людину з процесів опрацювання інформації, уникнути втрат істотної інформації і мінімізувати виникнення помилок на ранніх етапах життєвого циклу ПЗ

Ключові слова: специфікація вимог до програмного забезпечення (ПЗ), нефункційні характеристики ПЗ, інтелектуальний агент на основі онтологічного підходу

1. Вступ

На сьогодні людство все частіше покладається на програмне забезпечення (ПЗ) при вирішенні складних задач, стрімко зростає кількість програмних проектів з високою вартістю. Але, як показує статистика [1], частка проблемних програмних проектів (з перевитратами часу або коштів або з недостатнім функціоналом [1]) складає порядку половини всіх програмних проектів. Частка провальних програмних проектів (які скасовуються до завершення і ніколи не використовуються [1]) складає порядку 1/5 всіх програмних проектів. Отже, успішними програмними проектами, на які можна покладатись і на які варто витратити кошти, є лише 1/3 всіх програмних проектів.

Значна кількість помилок вноситься у ПЗ на початкових етапах життєвого циклу ПЗ. Переважна більшість аварій, пов'язаних із ПЗ, виникли через помилки у специфікації вимог [2]. Програмні проекти, специфікація вимог яких містить недостатню, неточну, неповну та суперечливу інформацію, не можуть мати успішної реалізації [3]. Крім цього, чим раніше буде виявлено дефект (помилку, порушення, недолік, несправність), тим дешевше обійдеться його виправлення. Тому для забезпечення необхідної функціональності та якості ПЗ необхідно здійснити дослідження специфікації вимог. Метою такого дослідження є виявлення та усунення недоліків початкових етапів життєвого циклу ПЗ та фактів недостатності інформації, котра має до них відношення. Хоча повнота вимог до ПЗ є бажаною, визначення повноти набору вимог не є реалістичним, що було доведено у [4]. Отже, в процесі такого дослідження необхідно оцінити, наскільки достатньо у специфікації відображена інформація щодо майбутнього ПЗ. Особливої уваги потребують вимоги, які характеризують нефункційні характеристики ПЗ [5].

Отже, сьогодні актуальною задачею є автоматизований аналіз специфікацій вимог до ПЗ на предмет достатності інформації щодо нефункційних характеристик-складових якості ПЗ. Згідно ISO 25010 [6], нефункційними характеристиками-складовими якості ПЗ є: надійність, функційна придатність, ефективність, сумісність, супроводжуваність, можливість переносу, захищеність, зручність використання.

2. Аналіз літературних даних та постановка проблеми

Зараз, під час розвитку епохи семантичного вебу, ключовою технологією є онтології. Вперше поняття онтології в галузі інформаційних технологій застосував Том Грубер [7]. Онтології використовуються для відображення відомих знань, а також надбання, структурування знань і формування нових знань предметної галузі. Перевагами використання онтологій є: системний підхід до вивчення предметної галузі, можливість цілісного подання відомої інформації предметної галузі, виявлення дублювань та прогалів у знаннях на основі візуалізації відсутніх логічних зв'язків. Крім цього, саме онтології забезпечують можливість автоматичного розуміння та аналізу інформації, усунення людини та мінімізацію інформаційних втрат в процесі опрацювання інформації та здобуття знань [7, 8].

Використання онтологій для інженерії ПЗ забезпечує уникнення повторної концептуалізації предметної галузі, скорочує використання ресурсів на ранніх етапах життєвого циклу ПЗ, зменшує кількість помилок концептуалізації [7–9].

Для підвищення достовірності оцінювання нефункційних характеристик-складових якості ПЗ значну цінність мають знання фахівців, що вже володіють таким досвідом. Наприклад, цінними є знання щодо взаємовпливу і кореляції нефункційних характеристик за атрибутами, оскільки частина атрибутів, за якими відбувається кореляція, може упускатись взагалі. Внаслідок цього може погіршуватись точність та достовірність отриманих оцінок і т. і. Пом'якшення впливу взаємної кореляції здійснюється шляхом виявлення спільних атрибутів, забезпечення їх наявності у специфікації вимог до ПЗ, підвищення точності їх значень. Отже, інформацію щодо оцінювання нефункційних характеристик-складових якості ПЗ (наприклад, взаємозв'язків нефункційних характеристик за атрибутами) зручно подавати у вигляді онтологій, які дають змогу відобразити причинно-наслідкові зв'язки між поняттями.

Чимало досліджень присвячені ідеї використання онтологій для галузі інженерії ПЗ. Так, методи та засоби побудови програмних систем на основі онтологічних моделей задач запропоновані у [8, 9]. Автори [8, 9] пропонують використовувати онтологічні моделі на всіх етапах життєвого циклу ПЗ за рахунок концептуалізації предметної галузі в контексті розв'язуваних задач. Тобто у предметній галузі виділяються необхідні сутності, визначаються їх атрибути та обмеження, залежності між ними і будується онтологія предметної галузі, яка надалі може використовуватись розробниками. Авторами [8, 9] доведено, що використання такої онтології предметної галузі спрощує процес початкової концептуалізації при створенні ПЗ, дозволяє уникнути помилок концептуалізації на ранніх етапах життєвого циклу ПЗ. Але використання такого підходу є неможливим, якщо ПЗ розробляється для предметної галузі, для якої ще не розроблено онтологію. Крім цього, при розробленні ПЗ слід враховувати вимоги та стандарти не лише предметної галузі, але й стандарти щодо розроблення ПЗ, що жодним чином не враховується в описаному у [8, 9] підході. Також методи та засоби побудови програмних систем на основі онтологічних моделей задач не дають можливості автоматизовано аналізувати специфікації вимог до ПЗ, зокрема, на предмет достатності їх інформації, хоча саме автоматизоване опрацювання знань забезпечує мінімізацію інформаційних втрат.

Підходи до трасування вимог до ПЗ на основі зважених онтологій розроблено в [10, 11]. Автори [10] пропонують онтологічну структуру трасованості багатocільових вимог MUPRET для опрацювання неоднорідності вимог до ПЗ на основі автоматичної генерації відношень трасованості. Точність відношень трасованості, що генеруються структурою MUPRET, перевіряються шляхом порівняння з набором відношень трасованості, які вручну ідентифіковані користувачами. В роботі [11] зважені онтології використовуються для опрацювання природної мови при переході від специфікації вимог, написаній природною мовою, до проектування ПЗ. Саме онтології, як доводять автори [11], здатні проявити невідповідності вимог, представлених природною мовою. Представлені підходи спрямовані на усунення неоднорідності або

невідповідності наявних вимог у специфікаціях, але жодним чином не перевіряють, чи всі потреби та вимоги користувача були відображені у специфікаціях. Отже, вони непридатні для оцінювання достатності інформації щодо нефункційних характеристик у специфікаціях вимог до ПЗ.

Онтологічну модель для опису та визначення предметних і операційних знань щодо забезпечення якості ПЗ розроблено у [12]. Автори [12] запропонували онтологію, яка об'єднує знання щодо термінології та семантичних відносин стандартів SWEBOOK, IEEE, ISO, спрямованих на забезпечення якості ПЗ. Розроблена онтологія моделює процес життєвого циклу ПЗ – забезпечення якості ПЗ. Але забезпечення якості відбувається лише в кінці життєвого циклу ПЗ для готового програмного коду. Сьогодні якість ПЗ трактується як його здатність задовольнити потреби замовника при використанні за певних умов [6]. Тому вся необхідна інформація щодо потреб замовника повинна бути закладена вже у специфікації вимог до ПЗ, тобто вже на основі специфікації можна оцінити достатність інформації для подальшого досягнення якості ПЗ. Підхід, запропонований у [12], не передбачає забезпечення якості ПЗ на ранніх етапах життєвого циклу, зокрема, не передбачає оцінювання достатності інформації щодо нефункційних характеристик-складових якості ПЗ у специфікаціях вимог до ПЗ.

Онтології та зважені онтології предметної галузі «Інженерія програмного забезпечення» (частини «Якість ПЗ», «Якість ПЗ. Метричний аналіз», «Специфікація вимог до ПЗ») розроблені у [13]. Вони є читабельними та адаптивними, оскільки представляють широко використовуваний для різних задач стандарт ISO 25010:2011. Це онтологічні моделі багаторазового використання, релевантні для вирішення задачі автоматизованого аналізу специфікацій вимог до ПЗ на предмет достатності інформації з метою підвищення якості розроблюваного за специфікаціями ПЗ. Тому зазначені онтології використовуватимуться для реалізації та роботи розроблюваного інтелектуального агента.

Онтології забезпечують можливість доступу, розуміння та аналізу інформації інтелектуальними агентами. Інтелектуальний агент використовує під час свого функціонування інформацію, отриману з навколишнього середовища, аналізує її, зіставляючи з уже відомими йому фактами і, на основі результатів аналізу, приймає рішення про подальші дії [14]. Саме в формуванні нових знань, зокрема, певних висновків та рекомендацій, і полягає інтелектуальність агента.

Ряд досліджень присвячено розробленню інтелектуальних агентів на основі онтологічного підходу для галузі інженерії ПЗ. Так, автори [15] досліджують застосування онтологій для агентно-орієнтованої програмної інженерії та наводять експериментальне підтвердження переваг застосування онтологічних агентів для галузі інженерії ПЗ. В результаті такого дослідження в [15] запропоновано агент, який використовує існуючі онтологічні конструкції для створення програмного коду на основі специфікацій. Недоліком такого агента є те, що він не виконує верифікацію та валідацію вимог специфікацій потребам замовників, не оцінює достатності наявної у специфікації вимог інформації, не виявляє потреб, які не були відображені у вимогах (зокрема, нефункційних).

Використання запропонованого агента може призвести до розроблення коректного ПЗ, яке, проте, не буде якісним внаслідок невідповідності всім потребам замовників.

Робота [16] присвячена усуненню невизначеності у вимогах до ПЗ та покращенню спілкування зацікавлених сторін шляхом впровадження інтелектуальних агентів на основі онтологічного підходу. Автори [17] пропонують розроблення інтелектуального агента на основі онтології для мінімізації існуючої семантичної невизначеності при розробленні специфікації вимог до ПЗ природньою (іспанською) мовою. Запропонований агент використовується також для автоматичного отримання основних елементів специфікації та для автоматичної побудови діаграми цілей. Представлені в роботах [16] та [17] агенти спрямовані на усунення невизначеності у наявних вимогах специфікацій, але не перевіряють, чи всі потреби користувача були відображені у таких вимогах, чи достатніми є вимоги специфікацій.

У роботі [18] онтологічні агентно-орієнтовані моделі використовуються для формалізації первинних вимог до ПЗ з метою зниження витрат на прикладі розроблення додатків для Ambient Assisted Living для пацієнтів із хворобою Паркінсона. Автори [19] запропонували основу для формального подання вимог обмежених ресурсами контекстних систем критичного застосування у вигляді інтелектуальних агентів на основі онтологічного підходу. Тобто автори робіт [18, 19] лише надають механізми на основі онтологій для формалізації вимог до ПЗ. Але зазначені підходи не аналізують вимог, не оцінюють достатності вимог у специфікації, не виявляють інформаційних втрат при формуванні вимог. А програмні проекти, специфікація вимог яких містить недостатню або неповну інформацію, не можуть мати успішної реалізації.

Автори [20] пропонують завдання-орієнтовану архітектуру на основі агентно-орієнтованої парадигми та онтологічного дизайну для систем підтримки прийняття рішень. Запропонована архітектура дозволяє ітеративно переносити функційні вимоги в архітектурні компоненти. Але ця архітектура жодним чином не працює з нефункційними вимогами – як не перевіряє їх на повноту і достатність, так і не переносить їх в архітектурні компоненти розроблюваного ПЗ. Отже, підхід, запропонований у [20], також є непридатним для автоматизованого аналізу специфікацій вимог до ПЗ на предмет достатності інформації щодо нефункційних характеристик-складових якості ПЗ.

Проведений аналіз відомих інтелектуальних агентів на основі онтологічного підходу показав, що відомі агенти спрямовані або на автоматизоване розроблення програмного коду за специфікацією, або на формалізацію наявних вимог, або ж, максимум, на усунення невизначеності у вимогах. Вони не розв'язують задачу оцінювання достатності інформації у вимогах до ПЗ на основі автоматизованого аналізу специфікацій вимог до ПЗ.

Щодо оцінювання достатності інформації вимог до ПЗ, то на сьогодні відомим є оцінювання достатності вимог щодо безпеки ПЗ [4, 21]. Автори [4, 21] надають метрики валідації, які використовують аналіз небезпек, а також похідні вимоги до ПЗ для пом'якшення виявлених небезпек з метою оцінювання достатності вимог безпеки ПЗ на початку процесу розроблення ПЗ. Зокрема,

в [4, 21] запропоновано нову модель для валідації вимог безпеки, зосереджуючись на достатності ідентифікації небезпек, аналізі небезпек і трасованості вимог безпеки. Крім цього, автори [4, 21] доводять, що будь-які метрики, що вимірюють глибину причинно-факторного аналізу ПЗ, повинні бути пристосовані до різних стандартів вимірювання достатності. В [4, 21] введено декілька різних метрик, зокрема, показник вимог безпеки як відношення кількості вимог безпеки до загальної кількості вимог. Даний показник автори пропонують порівнювати з аналогічним показником для іншого, вже реалізованого, ПЗ, на основі чого й робиться висновок про достатність вимог безпеки. Очевидно, що на основі такої метрики можна оцінити лише достатність кількості вимог безпеки, але аж ніяк не достатність їх інформації, адже недобросовісні розробники можуть штучно збільшити кількість вимог, наприклад, їх дублюванням. Ще одним недоліком такого підходу є неможливість інтерпретації шляхом порівняння запропонованого показника для принципово нового ПЗ. Крім цього, такий підхід не є автоматизованим, адже авторами не було запропоновано інструмента для автоматичної ідентифікації та підрахунку вимог безпеки у специфікації вимог до ПЗ.

Ще одним підходом до оцінювання достатності інформації в процесі розроблення ПЗ є оцінювання достатності тестування як досягнення рівнів тестового покриття, рекомендованих або затверджених стандартами безпеки та галузевими рекомендаціями [22]. В цьому дослідженні запропоновано емпіричну оцінку достатності тестування для систем бортового ПЗ з врахуванням верифікації ПЗ до поставлених вимог. Але цей підхід спрямований лише на верифікацію ПЗ та вимог, а не на валідацію розробленого ПЗ та потреб замовників. Крім цього, зазначений підхід використовує специфікацію вимог виключно як вхід для розробленого інструменту, але не перевіряє на предмет достатності вимоги специфікації.

Одним з рішень в галузі оцінювання достатності інформації у специфікаціях вимог до ПЗ є роботи [5, 13], в яких запропоновано теоретичні та прикладні засади інформаційної технології оцінювання достатності інформації щодо якості у специфікаціях вимог до ПЗ. Розроблені у [5, 13] засади спрямовані на визначення достатності інформації щодо якості у вимогах до ПЗ, але у цих роботах не реалізовано автоматизації такого оцінювання. Тому, саме підхід, запропонований у [5, 13], і буде розвинутий при подальшому розв'язанні задачі автоматизованого аналізу специфікацій вимог до ПЗ на предмет достатності інформації щодо нефункційних характеристик-складових якості ПЗ.

3. Мета та задачі дослідження

Метою дослідження є автоматизація аналізу специфікацій вимог до ПЗ на предмет достатності інформації та підвищення рівня достатності інформації щодо нефункційних характеристик-складових якості ПЗ у специфікаціях вимог. Досягнення цієї мети дозволить частково усунути людину з процесів опрацювання інформації та здобуття знань, уникнути втрат істотної інформації і мінімізувати виникнення помилок на ранніх етапах життєвого циклу.

Для досягнення цієї мети, потрібно розв'язати наступні задачі:

– розробити модель діяльності та реалізувати інтелектуальний агент на основі онтологічного підходу для оцінювання інформації про нефункційні характеристики у специфікації вимог до ПЗ (агент буде інтелектуальним, оскільки надаватиме нові знання – зокрема, висновки про достатність інформації та рекомендації щодо її підвищення);

– проаналізувати інформацію про нефункційні характеристики у специфікації вимог до ПЗ з використанням розробленого інтелектуального агента на основі онтологічного підходу.

4. Модель діяльності інтелектуального агента для аналізу нефункційних характеристик у специфікації вимог до програмного забезпечення

Пропонований інтелектуальний агент на основі онтологічного підходу використовує під час свого функціонування базові онтології нефункційних характеристик-складових якості ПЗ (які були розроблені у [5, 13]) як відомі йому факти. Саме онтології, які відображають причинно-наслідкові зв'язки між поняттями, дозволяють виявити відсутні у специфікації атрибути і встановити, які нефункційні характеристики-складові якості ПЗ неможливо визначити без таких атрибутів. Наприклад, фрагмент базової онтології для Сумісності (Compatibility), реалізованої в Protégé 4.2, представлений на рис. 1. З цими базовими онтологіями агент зіставляє інформацію, отриману зі специфікації вимог до реального ПЗ, представлену у вигляді реальних онтологій для забезпечення можливості порівняння базових та реальних онтологій. На основі такого порівняння онтологій, виконаного в Protégé 4.2, інтелектуальний агент отримує список відсутніх у специфікації атрибутів, оскільки саме відсутніми у специфікації атрибутами відрізнятимуться реальні онтології від базових. Інтелектуальний агент проводить аналіз отриманого списку відсутніх атрибутів та залежностей нефункційних характеристик від атрибутів (за базовими онтологіями) і встановлює, які нефункційні характеристики-складові якості ПЗ неможливо визначити без відсутніх атрибутів. Крім цього, інтелектуальний агент підраховує кількості відсутніх атрибутів та нефункційних характеристик, які неможливо обчислити без певних атрибутів, для формування числової оцінки достатності інформації у специфікаціях вимог до ПЗ. Після цього інтелектуальний агент оцінює інформацію у специфікації вимог до ПЗ та приймає рішення про подальші дії, зокрема, надає висновки про достатність інформації та рекомендації щодо її підвищення.

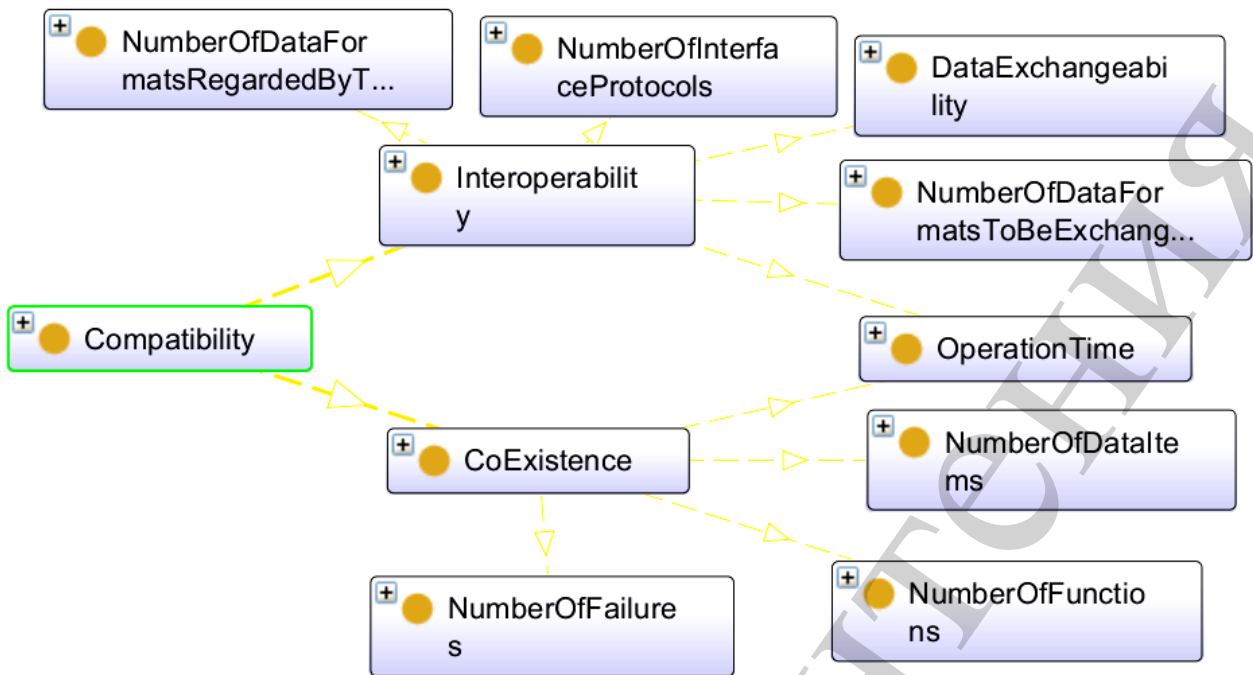


Рис. 1. Базова онтологія для Сумісності (Compatibility)

Отже, процес оцінювання специфікації вимог інтелектуальним агентом полягає у:

1) порівнянні онтологій нефункційних характеристик-складових якості реального ПЗ з базовими онтологіями нефункційних характеристик-складових якості ПЗ з метою виявлення атрибутів, відсутніх у специфікації вимог до реального ПЗ, за якою реальні онтології були побудовані;

2) виявленні підхарактеристик та нефункційних характеристик-складових якості ПЗ, які неможливо обчислити на основі наявних у специфікації вимог до реального ПЗ атрибутів;

3) формуванні висновку про достатність або недостатність інформації у специфікації вимог для визначення кожної нефункційної характеристики ПЗ окремо та для визначення всіх нефункційних характеристик ПЗ разом;

4) розрахунку числових оцінок рівня достатності наявної у специфікації вимог інформації для визначення кожної нефункційної характеристики ПЗ за формулою (1):

$$D_j = \frac{\left(k_j - \sum_{i=1}^{k_j} \frac{qm_i}{qn_i} \right)}{k_j}, \quad (1)$$

де k_j – кількість підхарактеристик j -ї нефункційної характеристики ПЗ ($j=1\dots 8$, оскільки у стандарті ISO 25010 [6] визначено саме 8 нефункційних характеристик-складових якості ПЗ), qm_i – кількість відсутніх у специфікації вимог до реального ПЗ атрибутів для i -ї підхарактеристики j -ї нефункційної характеристики ПЗ, qn_i – кількість необхідних атрибутів для i -ї

підхарактеристики j -ї нефункційної характеристики ПЗ (визначається базовими онтологіями для кожної нефункційної характеристики-складової якості ПЗ);

5) розрахунку числової оцінки рівня достатності наявної у специфікації вимог інформації для визначення всіх нефункційних характеристик-складових якості ПЗ за формулою (2):

$$D = \frac{\left(k - \sum_{j=1}^k \frac{qmc_j}{qnc_j} \right)}{k}, \quad (2)$$

де k – кількість нефункційних характеристик–складових якості ПЗ ($k=8$ відповідно до ISO 25010 [6]), qmc_j – кількість відсутніх у специфікації вимог до реального ПЗ атрибутів для j -ї нефункційної характеристики ПЗ, qnc_j – кількість необхідних атрибутів для j -ї нефункційної характеристики ПЗ (визначається базовими онтологіями для кожної нефункційної характеристики-складової якості ПЗ);

б) візуалізації прогалин у знаннях щодо нефункційних характеристик-складових якості ПЗ.

Пропонований агент є інтелектуальним, тому що він автоматично опрацьовує наявні знання (вимоги щодо нефункційних характеристик, представлені у вигляді онтологій) та формує нові знання (висновки про рівень достатності інформації, рекомендації щодо підвищення рівня достатності інформації у специфікації вимог).

Даний інтелектуальний агент не працює з нечіткими даними, оскільки задача оцінювання достатності інформації не передбачає нечіткості. Необхідний атрибут або присутній у специфікації, або відсутній у ній і тоді відбувається падіння рівня достатності інформації на величину, яка залежить від того, скільки нефункційних характеристик залежать від даного атрибуту (корелюють за ним).

Нехай:

$$SMM_{Fs} = O_{Fs} \setminus (O_{Fs} \cap O_{Fs_real}),$$

де $SMM_{Fs} = \{fsa_1, \dots, fsa_{(19-m)}\}$ – множина атрибутів підхарактеристик Функційної придатності (Functional Suitability), відсутніх у специфікації вимог до реального ПЗ, а також кількість підхарактеристик Функційної придатності (Functional Suitability), які неможливо обчислити на основі наявних атрибутів (необхідна кількість атрибутів визначається за стандартом ISO 25023:2016 [23]); O_{Fs} – базова онтологія Функційної придатності (Functional Suitability); O_{Fs_real} – онтологія Функційної придатності (Functional Suitability) для реального ПЗ;

$$SMM_{Pe} = O_{Pe} \setminus (O_{Pe} \cap O_{Pe_real}),$$

де $SMM_{Pe} = \{pea_1, \dots, pea_{(30-n)}\}$ – множина атрибутів підхарактеристик Ефективності (Performance Efficiency), відсутніх у специфікації вимог до реального ПЗ, а також кількість підхарактеристик Ефективності (Performance Efficiency), які неможливо обчислити на основі наявних атрибутів; O_{Pe} – базова онтологія Ефективності (Performance Efficiency); O_{Pe_real} – онтологія Ефективності (Performance Efficiency) для реального ПЗ;

$$SMM_{Ub} = O_{Ub} \setminus (O_{Ub} \cap O_{Ub_real}),$$

де $SMM_{Ub} = \{uba_1, \dots, uba_{(56-k)}\}$ – множина атрибутів підхарактеристик Зручності використання (Usability), відсутніх у специфікації вимог до реального ПЗ, а також кількість підхарактеристик Зручності використання (Usability), які неможливо обчислити на основі наявних атрибутів; O_{Ub} – базова онтологія Зручності використання (Usability); O_{Ub_real} – онтологія Зручності використання (Usability) для реального ПЗ;

$$SMM_{Rb} = O_{Rb} \setminus (O_{Rb} \cap O_{Rb_real}),$$

де $SMM_{Rb} = \{rba_1, \dots, rba_{(35-o)}\}$ – множина атрибутів підхарактеристик Надійності (Reliability), відсутніх у специфікації вимог до реального ПЗ, а також кількість підхарактеристик Надійності (Reliability), які неможливо обчислити на основі наявних атрибутів; O_{Rb} – базова онтологія Надійності (Reliability); O_{Rb_real} – онтологія Надійності (Reliability) для реального ПЗ;

$$SMM_{Cb} = O_{Cb} \setminus (O_{Cb} \cap O_{Cb_real}),$$

де $SMM_{Cb} = \{cba_1, \dots, cba_{(12-p)}\}$ – множина атрибутів підхарактеристик Сумісності (Compatibility), відсутніх у специфікації вимог до реального ПЗ, а також кількість підхарактеристик Сумісності (Compatibility), які неможливо обчислити на основі наявних атрибутів; O_{Cb} – базова онтологія Сумісності (Compatibility); O_{Cb_real} – онтологія Сумісності (Compatibility) для реального ПЗ;

$$SMM_{Scr} = O_{Scr} \setminus (O_{Scr} \cap O_{Scr_real}),$$

де $SMM_{Scr} = \{scra_1, \dots, scra_{(29-q)}\}$ – множина атрибутів підхарактеристик Захищеності (Security), відсутніх у специфікації вимог до реального ПЗ, а також кількість підхарактеристик захищеності, які неможливо обчислити на основі наявних атрибутів; O_{Scr} – базова онтологія Захищеності (Security); O_{Scr_real} – онтологія Захищеності (Security) для реального ПЗ;

$$SMM_{Mb} = O_{Mb} \setminus (O_{Mb} \cap O_{Mb_real}),$$

де $SMM_{Mb} = \{mba_1, \dots, mba_{(39-r)}\}$ – множина атрибутів підхарактеристик Супроводжуваності (Maintainability), відсутніх у специфікації вимог до реального ПЗ, а також кількість підхарактеристик Супроводжуваності (Maintainability), які неможливо обчислити на основі наявних атрибутів; O_{Mb} – базова онтологія Супроводжуваності (Maintainability); O_{Mb_real} – онтологія Супроводжуваності (Maintainability) для реального ПЗ;

$$SMM_{Pb} = O_{Pb} \setminus (O_{Pb} \cap O_{Pb_real}),$$

де $SMM_{Pb} = \{pba_1, \dots, pba_{(22-1)}\}$ – множина атрибутів підхарактеристик Можливості переносу (Portability), відсутніх у специфікації вимог до реального ПЗ, а також кількість підхарактеристик Можливості переносу (Portability), які неможливо обчислити на основі наявних атрибутів; O_{Pb} – базова онтологія Можливості переносу (Portability); O_{Pb_real} – онтологія Можливості переносу (Portability) для реального ПЗ.

Продукційні правила для формування висновку щодо достатності або недостатності інформації у специфікації вимог для визначення кожної нефункційної характеристики-складової якості ПЗ:

– для Функційної придатності (Functional Suitability):

*if $SMM_{Fs} = \emptyset$ then "SRS information is sufficient for Functional Suitability"
else "SRS information is insufficient for Functional Suitability";*

– для Ефективності (Performance Efficiency):

*if $SMM_{Pe} = \emptyset$ then "SRS information is sufficient for Performance Efficiency"
else "SRS information is insufficient for Performance Efficiency";*

– для Зручності використання (Usability):

*if $SMM_{Ub} = \emptyset$ then "SRS information is sufficient for Usability"
else "SRS information is insufficient for Usability";*

– для Надійності (Reliability):

*if $SMM_{Rb} = \emptyset$ then "SRS information is sufficient for Reliability"
else "SRS information is insufficient for Reliability";*

– для Сумісності (Compatibility):

*if $SMM_{cb} = \emptyset$ then "SRS information is sufficient for Compatibility"
else "SRS information is insufficient for Compatibility";*

– для Захищеності (Security):

*if $SMM_{scr} = \emptyset$ then "SRS information is sufficient for Security"
else "SRS information is insufficient for Security";*

– для Супроводжуваності (Maintainability):

*if $SMM_{mb} = \emptyset$ then "SRS information is sufficient for Maintainability"
else "SRS information is insufficient for Maintainability";*

– для Можливості переносу (Portability):

*if $SMM_{pb} = \emptyset$ then "SRS information is sufficient for Portability"
else "SRS information is insufficient for Portability".*

Продукційне правило для формування висновку щодо достатності або недостатності інформації у специфікації вимог для визначення всіх нефункційних характеристик-складових якості ПЗ:

*if $(SMM_{fs} \cup SMM_{pe} \cup SMM_{ub} \cup SMM_{rb} \cup SMM_{cb} \cup SMM_{scr} \cup SMM_{mb} \cup SMM_{pb}) = \emptyset$
then "SRS information is sufficient" else "SRS information is insufficient".*

Розроблена модель діяльності інтелектуального агента на основі онтологічного підходу для оцінювання інформації щодо нефункційних вимог у специфікаціях відображає особливості оцінювання достатності інформації для визначення нефункційних характеристик-складових якості ПЗ. Дана модель є теоретичним підґрунтям для реалізації інтелектуального агента на основі онтологічного підходу для оцінювання специфікацій вимог до ПЗ.

Метод діяльності інтелектуального агента на основі онтологічного підходу для оцінювання інформації щодо нефункційних вимог у специфікаціях вимог до ПЗ був розроблений у [24].

5. Аналіз нефункційних характеристик у специфікаціях вимог до програмного забезпечення за допомогою інтелектуального агента

На основі запропонованої моделі та розробленого методу був реалізований інтелектуальний агент на основі онтологічного підходу для оцінювання специфікацій вимог до ПЗ мовою РНР з використанням Protégé 4.2 (для роботи з онтологіями).

Аналіз інформації розділу «Специфічні вимоги» (згідно зі стандартом ISO 29148:2011 [25], саме цей розділ специфікації має підрозділ «Атрибути програмної системи», який може містити значення всіх атрибутів, необхідних для визначення нефункціональних характеристик-складових якості ПЗ) специфікації вимог до ПЗ транспортно-логістичної системи ТОВ «Гілея» (Україна), дав можливість побудувати онтології нефункційних характеристик для реального ПЗ. Відповідно до розроблених моделі та методу діяльності, реалізований інтелектуальний агент на основі онтологічного підходу виконує порівняння онтологій нефункційних характеристик для реального ПЗ з базовими онтологіями нефункційних характеристик-складових якості ПЗ.

В результаті такого порівняння, інтелектуальний агент на основі онтологічного підходу надав множину відсутніх атрибутів у специфікації вимог до ПЗ транспортно-логістичної системи ТОВ «Гілея»:

- для Зручності використання (Usability): Number of IO Data Items, Function Understandability, Number Of Tasks, Help Accessibility, Operation Time, Number of Screens or Forms, Number of Interface Elements, Number of Unsuccessfully Recovered Situation, Number of Functions Implemented with User Error Tolerance, Number of Interface Graphical Elements, Degree of Ergonomic Attractiveness, Freedom from Risk for Users with Specified Disabilities;

- для Функційної придатності (Functional Suitability): Functional Adequacy, Functional Implementation Completeness, Operation Time, Number of Data Items;

- для Ефективності (Performance Efficiency): Operation Time, Number of Tasks, Mean Amount of Throughput, Number of Failures, IO Loading Limits, Maximum Memory Utilization, Size of Database;

- для Надійності (Reliability): Operation Time, Number of Failures, Number of Resolved Failures, Number of Breakdowns;

- для Сумісності (Compatibility): Operation Time, Number of Failures, Number of Data Items, Data Exchangeability;

- для Захищеності (Security): Operation Time, Number of Data Items, Access Controllability, Number of Access Types, Number of Provided Authentication Methods, Number of Events Processed Using Digital Signature;

- для Супроводжуваності (Maintainability): Operation Time, Number of Failures, Number of Resolved Failures, Number of Variables, Number of Data Items;

- для Можливості переносу (Portability): Operation Time, Number of Data Items, Number of Data Structures.

Аналіз впливу кожного елемента множини відсутніх атрибутів на нефункційні характеристик та їх підхарактеристики, проведений реалізованим інтелектуальним агентом на основі онтологічного підходу, дав можливість обчислити кількості відсутніх атрибутів для підхарактеристик кожної нефункційної характеристики. Крім цього, такий аналіз дав можливість сформулювати висновок, що на основі наявних у специфікації вимог до реального ПЗ атрибутів неможливо обчислити такі підхарактеристики нефункційних характеристик, як:

- підхарактеристики Зручності використання (Usability): Appropriateness, Recognisability, Learnability, Operability, User Error Protection, User Interface

Aesthetics, Accessibility, тобто інформації в цій специфікації недостатньо для визначення всіх 6-и підхарактеристик Зручності використання (Usability);

– підхарактеристики Функційної придатності (Functional Suitability): Functional Completeness, Functional Correctness, Functional Appropriateness, тобто інформації в цій специфікації недостатньо для визначення всіх 3-х підхарактеристик Функційної придатності (Functional Suitability);

– підхарактеристики Ефективності (Performance Efficiency): Time Behaviour, Resource Utilization, Capacity, тобто інформації в цій специфікації недостатньо для визначення всіх 3-х підхарактеристик Ефективності (Performance Efficiency);

– підхарактеристики Надійності (Reliability): Maturity, Availability, Fault Tolerance, Recoverability, тобто інформації в цій специфікації недостатньо для визначення всіх 4-х підхарактеристик Надійності (Reliability);

– підхарактеристики Сумісності (Compatibility): CoExistence, Interoperability, тобто інформації в цій специфікації недостатньо для визначення обох підхарактеристик Сумісності (Compatibility);

– підхарактеристики Захищеності (Security): Confidentiality, Integrity, NonRepudiation, Authenticity, тобто інформації в цій специфікації недостатньо для визначення 4-х (з 5-и) підхарактеристик Захищеності (Security);

– підхарактеристики Супроводжуваності (Maintainability): Modularity, Analysability, Modifiability, Testability, тобто інформації в цій специфікації недостатньо для визначення 4-х (з 5-и) підхарактеристик Супроводжуваності (Maintainability);

– підхарактеристики Можливості переносу (Portability): Adaptability, Replaceability, тобто інформації в цій специфікації недостатньо для визначення 2-х (з 3-х) підхарактеристик Можливості переносу (Portability).

На основі проведеного аналізу, реалізований інтелектуальний агент сформував висновок про недостатність інформації у специфікації до ПЗ транспортно-логістичної системи ТОВ «Гілея» для визначення всіх нефункційних характеристик. Числові оцінки рівня достатності наявної у цій специфікації вимог інформації становлять:

– для визначення Usability:

$$D_{Ub} = \frac{\left(6 - \left(\frac{2}{6} + \frac{3}{8} + \frac{3}{13} + \frac{2}{11} + \frac{3}{6} + \frac{1}{5}\right)\right)}{6} = 0,70;$$

– для визначення Functional Suitability:

$$D_{Fs} = \frac{\left(3 - \left(\frac{2}{4} + \frac{3}{5} + \frac{3}{6}\right)\right)}{3} = 0,47;$$

– для визначення Performance Efficiency:

$$D_{Pe} = \frac{\left(3 - \left(\frac{3}{7} + \frac{4}{14} + \frac{3}{5}\right)\right)}{3} = 0,56;$$

– для визначення Reliability:

$$D_{Rb} = \frac{\left(4 - \left(\frac{3}{14} + \frac{1}{4} + \frac{2}{5} + \frac{2}{7}\right)\right)}{4} = 0,71;$$

– для визначення Compatibility:

$$D_{Cb} = \frac{\left(2 - \left(\frac{3}{4} + \frac{2}{5}\right)\right)}{2} = 0,43;$$

– для визначення Security:

$$D_{Scr} = \frac{\left(5 - \left(\frac{4}{10} + \frac{4}{8} + \frac{1}{2} + \frac{0}{2} + \frac{1}{1}\right)\right)}{5} = 0,52;$$

– для визначення Maintainability:

$$D_{Mb} = \frac{\left(5 - \left(\frac{4}{7} + \frac{0}{6} + \frac{2}{6} + \frac{2}{8} + \frac{2}{6}\right)\right)}{5} = 0,70;$$

– для визначення Portability:

$$D_{Pb} = \frac{\left(3 - \left(\frac{3}{11} + \frac{0}{4} + \frac{1}{3}\right)\right)}{3} = 0,80.$$

Числова оцінка рівня достатності наявної у специфікації вимог до ПЗ транспортно-логістичної системи ТОВ «Гілея» інформації для визначення всіх нефункційних характеристик-складових якості ПЗ становить:

$$D = \frac{\left(8 - \left(\frac{14}{49} + \frac{8}{15} + \frac{10}{26} + \frac{8}{30} + \frac{5}{9} + \frac{10}{23} + \frac{10}{33} + \frac{4}{18} \right) \right)}{8} = 0,63.$$

Так, реалізований інтелектуальний агент надає наступний висновок: «Наявних атрибутів в аналізованій специфікації недостатньо для визначення всіх нефункційних характеристик. Рівні достатності інформації в аналізованій специфікації: 70 % для Зручності використання (Usability); 47 % для Функційної придатності (Functional Suitability); 56 % для Ефективності (Performance Efficiency); 71 % для Надійності (Reliability); 43 % для Сумісності (Compatibility); 52 % для Захищеності (Security); 70 % для Супроводжуваності (Maintainability); 80 % для Можливості переносу (Portability); 63 % для всіх нефункційних характеристик разом. Є потреба в доповненні цієї специфікації атрибутами для всіх нефункційних характеристик».

Інтелектуальний агент надає також візуалізацію прогалин у знаннях про всі нефункційні характеристики. Така візуалізація представлена на рис. 2–9, де відсутні атрибути позначені як викреслені з базової онтології, реалізованої в Protégé 4.2, а підхарактеристики, для визначення яких недостатньо наявних атрибутів, окреслені колом у відповідній базовій онтології.

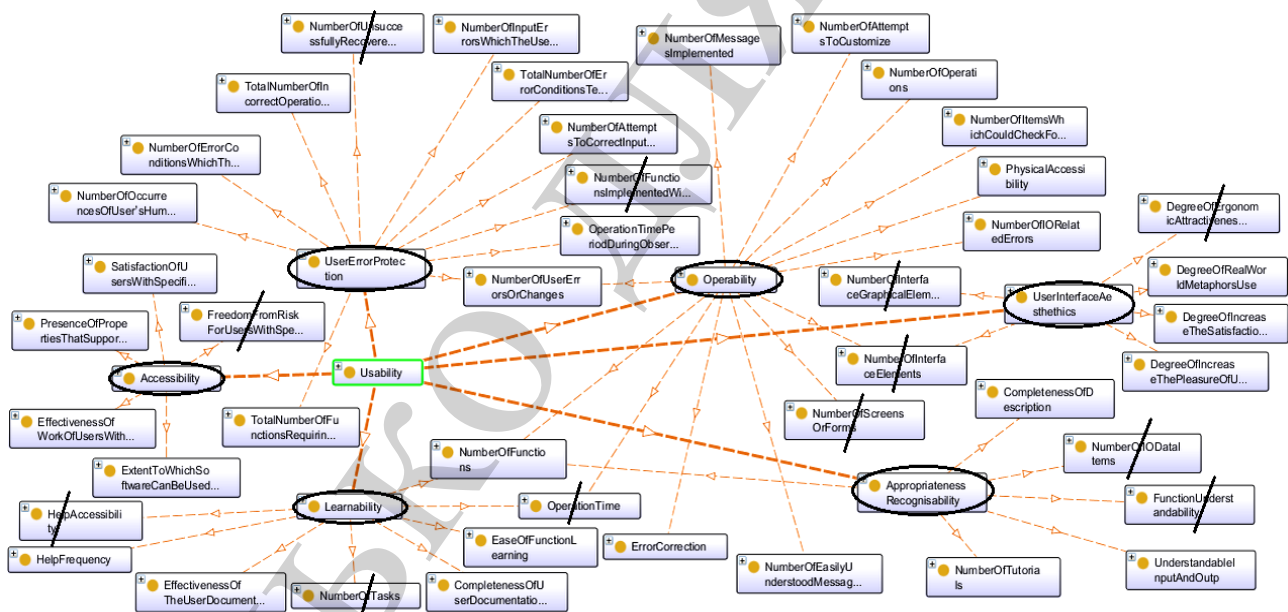


Рис. 2. Візуалізація прогалин у знаннях щодо Зручності використання (Usability)

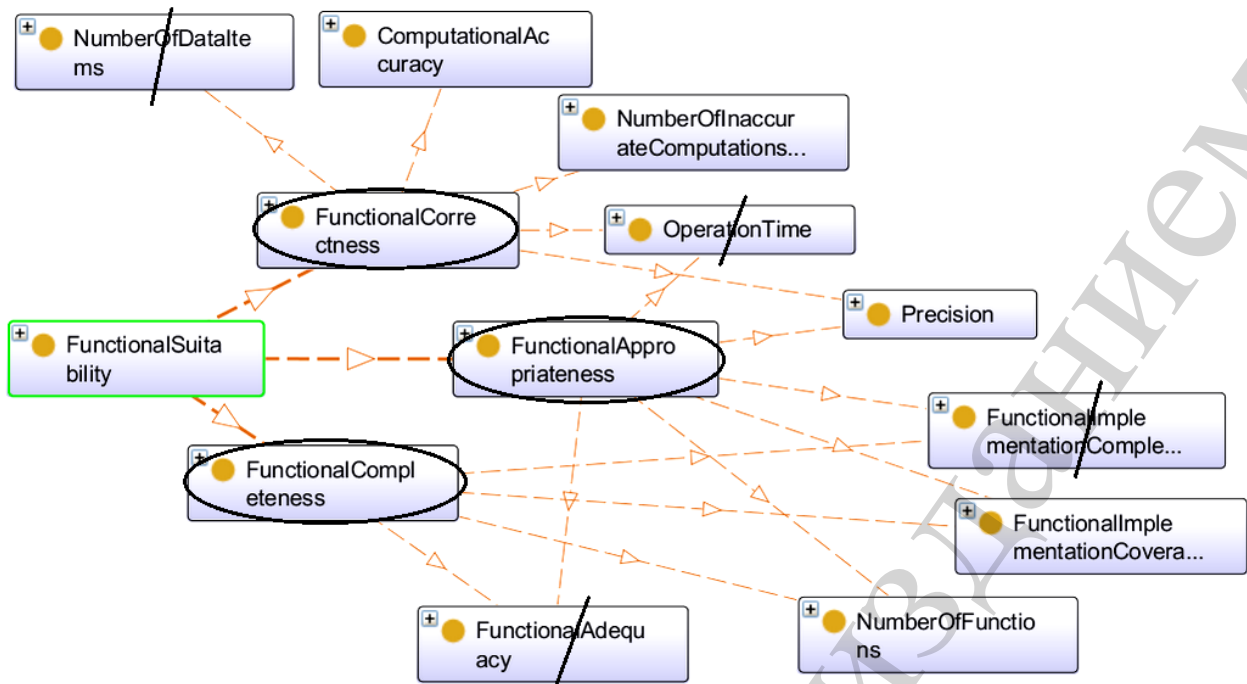


Рис. 3. Візуалізація прогалин у знаннях щодо Функційної придатності (Functional Suitability)

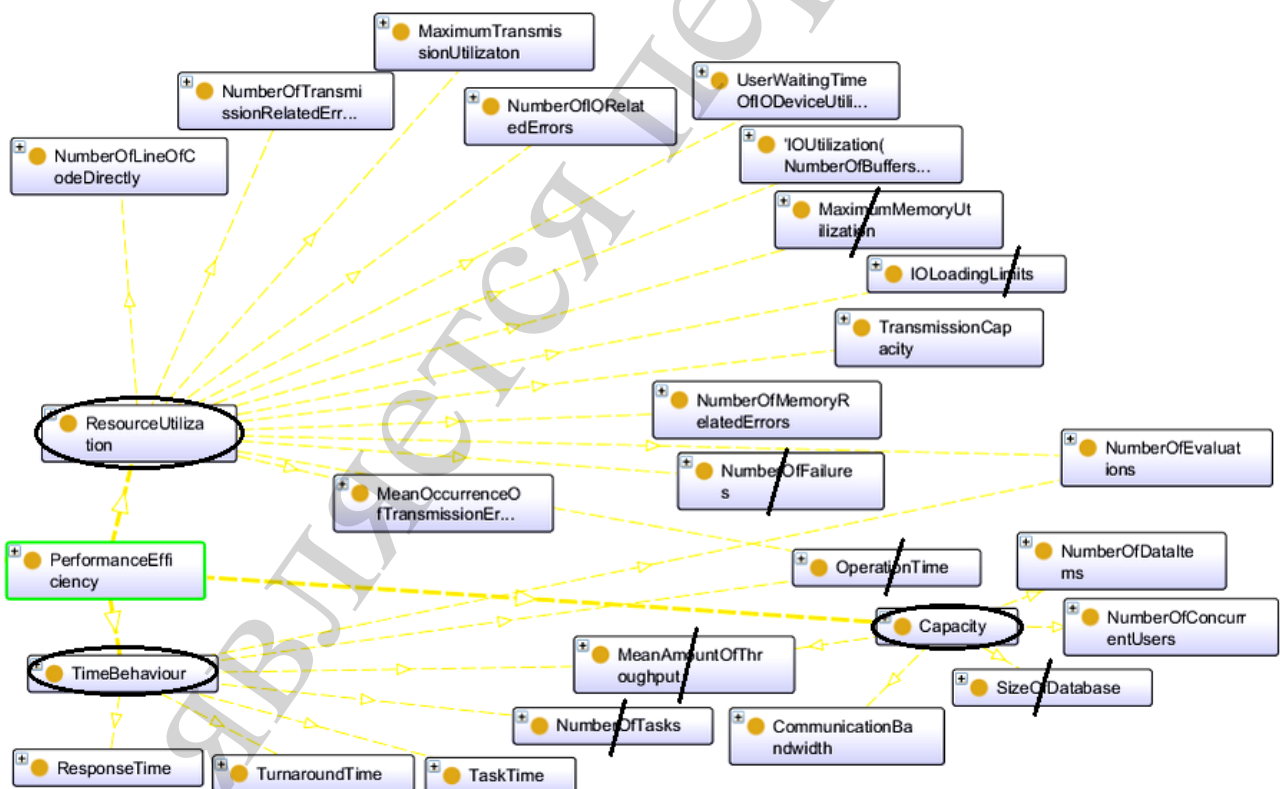


Рис. 4. Візуалізація прогалин у знаннях щодо Ефективності (Performance Efficiency)

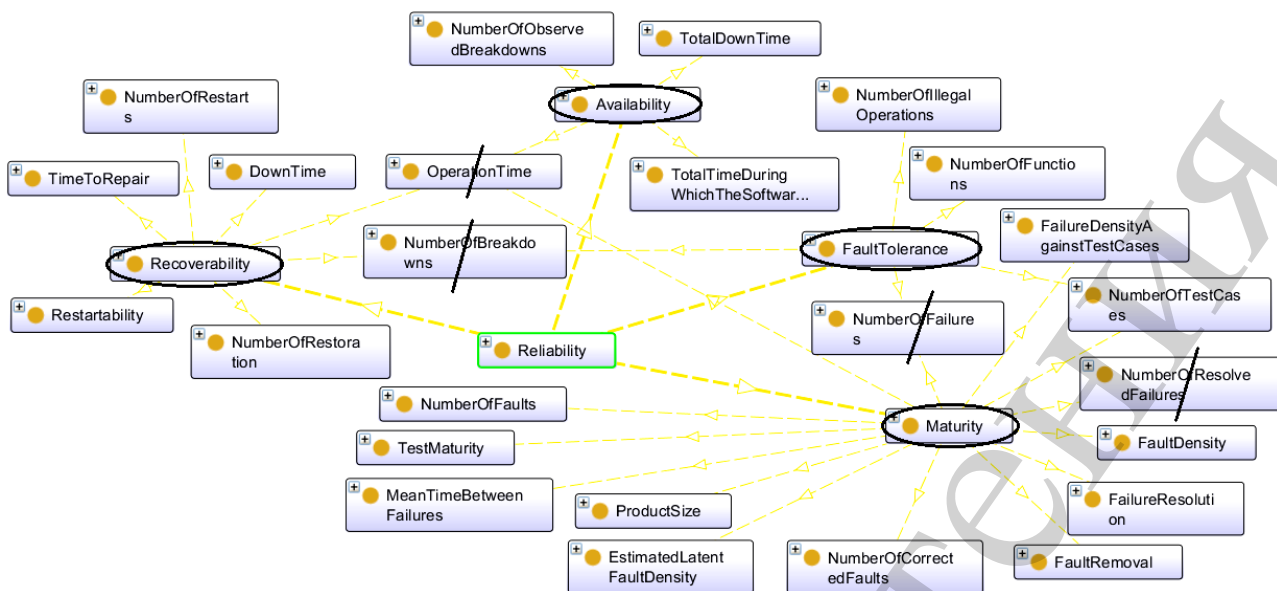


Рис. 5. Візуалізація прогалин у знаннях щодо Надійності (Reliability)

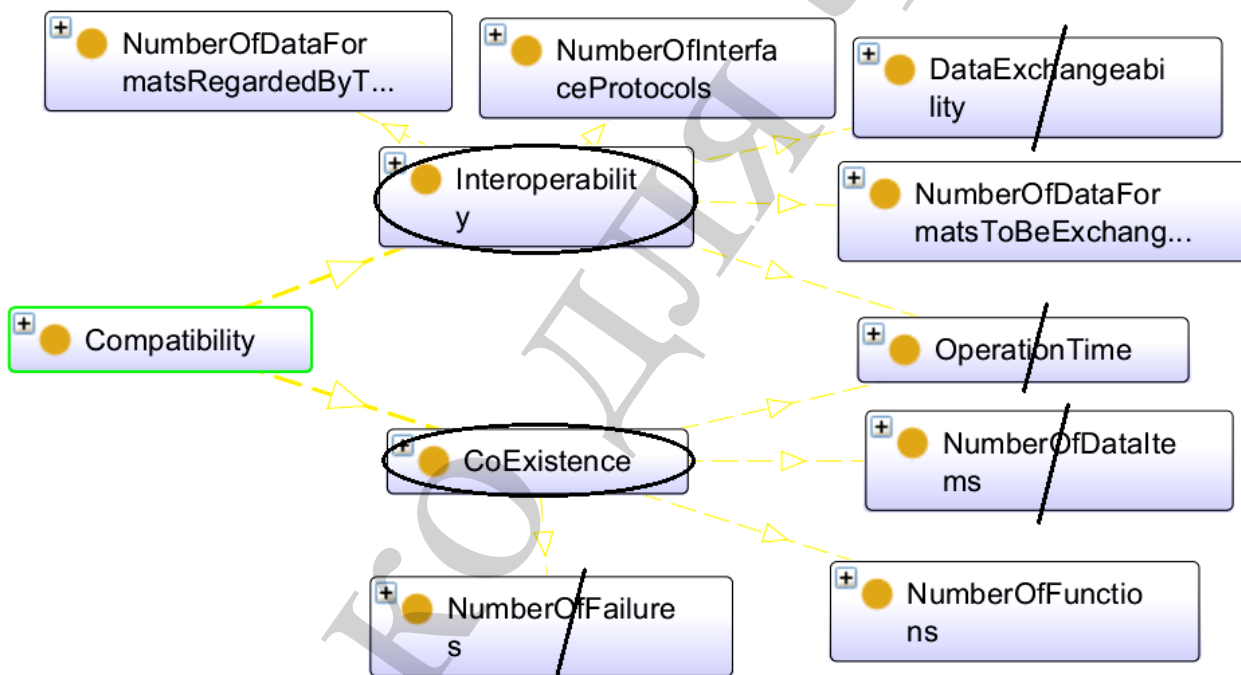


Рис. 6. Візуалізація прогалин у знаннях щодо Сумісності (Compatibility)

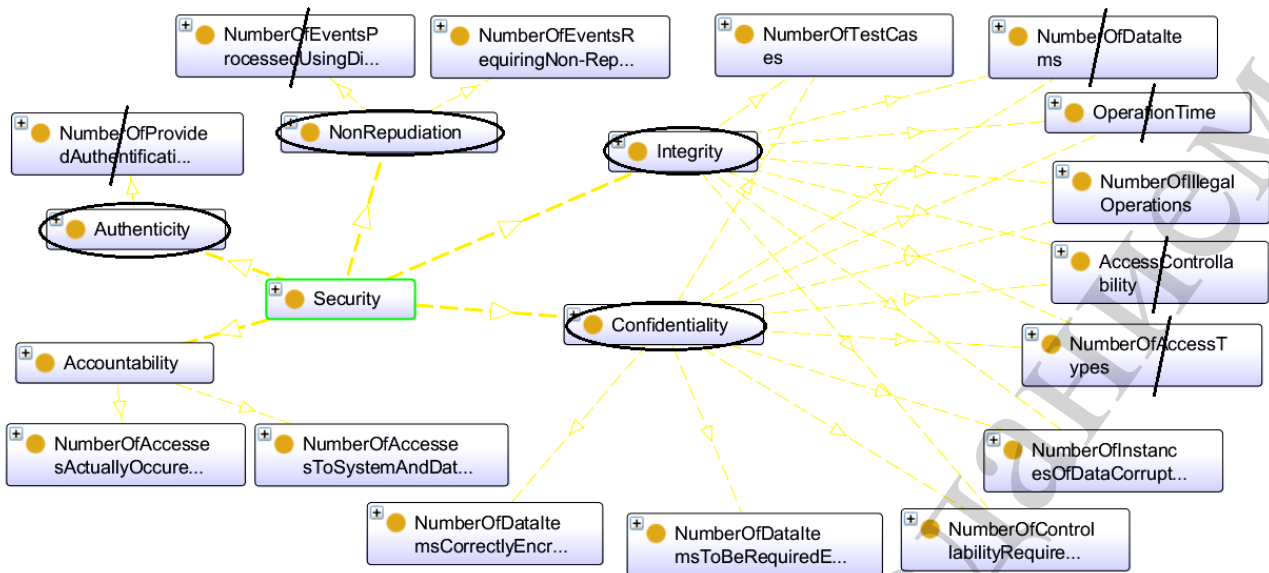


Рис. 7. Візуалізація прогалін у знаннях щодо Захищеності (Security)

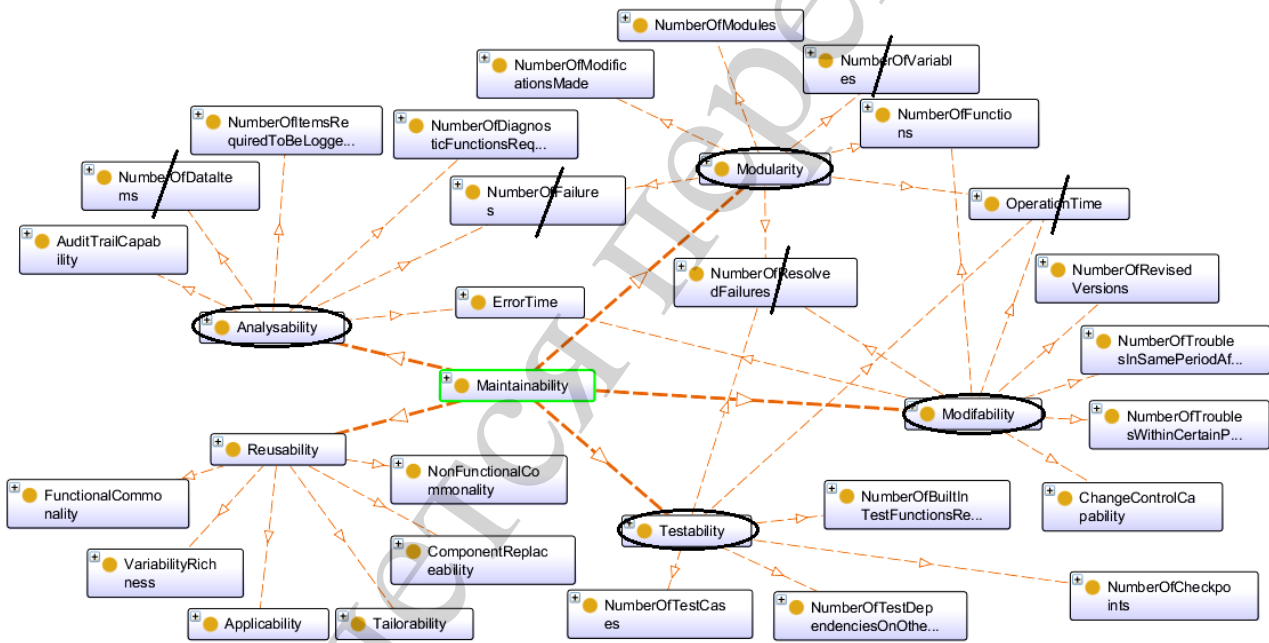


Рис. 8. Візуалізація прогалін у знаннях щодо Супроводжуваності (Maintainability)

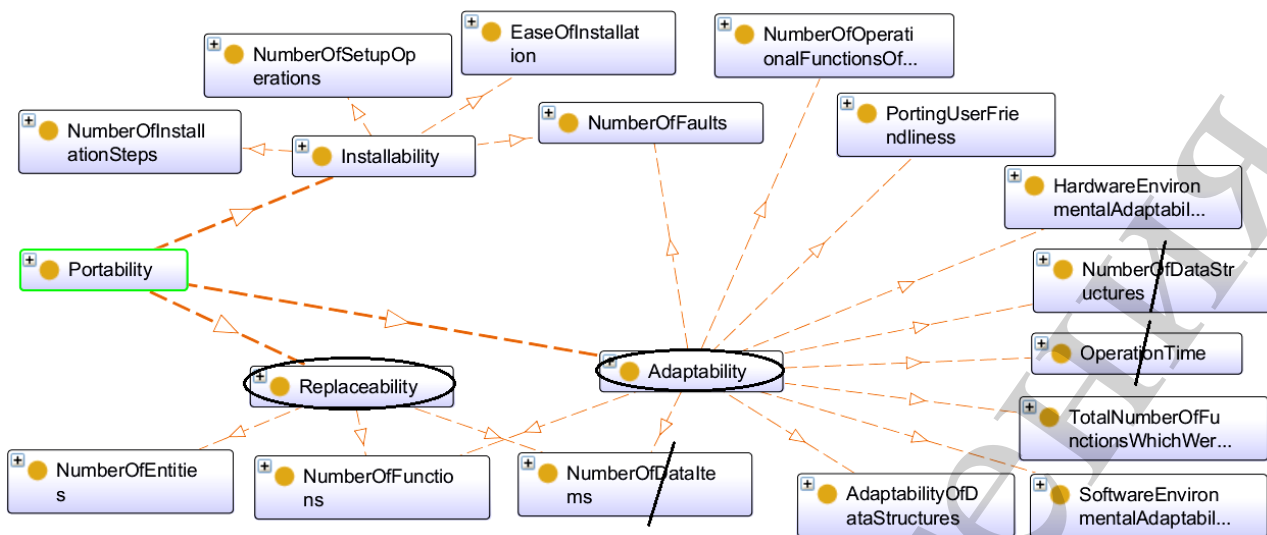


Рис. 9. Візуалізація прогалін у знаннях щодо Можливості переносу (Portability)

Для проведення другого експерименту використовувалась специфікація вимог до реального ПЗ автоматизованої системи управління виробничими процесами ТОВ «Деймос» (Україна). Аналіз інформації розділу «Специфічні вимоги» цієї специфікації дав можливість побудувати онтології нефункційних характеристик для реального ПЗ. Відповідно до розроблених моделі та методу діяльності, реалізований інтелектуальний агент виконав порівняння реальних онтологій з базовими онтологіями нефункційних характеристик-складових якості ПЗ. В результаті такого порівняння, інтелектуальний агент на основі онтологічного підходу надав наступну множину відсутніх атрибутів у специфікації вимог до ПЗ автоматизованої системи управління виробничими процесами ТОВ «Деймос»:

- для Зручності використання (Usability): Number of Functions, Number of IO Data Items, Number of Tutorials, Number of Tasks, Completeness of User Documentation and/or Help Facility, Number of Screens or Forms, Number of IO Related Errors, Number of Interface Elements, Number of Easily Understood Messages, Total Number of Error Conditions Tested, Total Number of Incorrect Operation Patterns, Number of Interface Graphical Elements, Degree of Ergonomic Attractiveness, Satisfaction of Users with Specified Disabilities;

- для Функційної придатності (Functional Suitability): Number of Functions, Functional Adequacy, Number of Data Items, Computational Accuracy;

- для Ефективності (Performance Efficiency): Number of Tasks, Number of Evaluations, Number of Failures, Number of IO Related Errors, Number of Data Items, Number of Memory Related Errors, Maximum Memory Utilization, Size of Database;

- для Надійності (Reliability): Number of Failures, Number of Test Cases, Number of Resolved Failures, Failure Density Against Test Cases, Number of Observed Breakdowns, Number of Functions, Number of Breakdowns, Restartability;

– для Сумісності (Compatibility): Number of Failures, Number of Functions, Number of Data Items, Number of Data Formats To Be Exchanged, Number of Interface Protocols;

– для Захищеності (Security): Number of Test Cases, Number of Data Items, Number of Access Types, Number of Data Items To Be Required Encryption/Decryption, Number of Events Processed Using Digital Signature, Number of Accesses to System and Data Recorded in the System Log, Number of Provided Authentication Methods;

– для Супроводжуваності (Maintainability): Number of Failures, Number of Resolved Failures, Number of Functions, Number of Modules, Variability Richness, Component Replaceability, Number of Data Items, Number of Diagnostic Functions Required, Number of Test Cases, Number of Checkpoints;

– для Можливості переносу (Portability): Number of Functions, Number of Data Items, Number of Data Structures, Number of Setup Operations, Number of Installation Steps.

Аналіз впливу кожного елемента множини відсутніх атрибутів на нефункційні характеристик та їх підхарактеристики, проведений реалізованим інтелектуальним агентом на основі онтологічного підходу, дав можливість обчислити кількості відсутніх атрибутів для підхарактеристик кожної нефункційної характеристики. Крім цього, такий аналіз дав можливість сформулювати висновок, що на основі наявних у специфікації вимог до реального ПЗ атрибутів неможливо обчислити всі підхарактеристики нефункційних характеристик.

На основі проведеного аналізу, *реалізований інтелектуальний агент надає наступний висновок*: «Наявних атрибутів в аналізованій специфікації недостатньо для визначення всіх нефункційних характеристик. Рівні достатності інформації в аналізованій специфікації: 64 % для Зручності використання (Usability); 59 % для Функційної придатності (Functional Suitability); 65 % для Ефективності (Performance Efficiency); 60 % для Надійності (Reliability); 43 % для Сумісності (Compatibility); 45 % для Захищеності (Security); 57 % для Супроводжуваності (Maintainability); 52 % для Можливості переносу (Portability); 59 % для всіх нефункційних характеристик разом. Є потреба в доповненні цієї специфікації атрибутами для всіх нефункційних характеристик».

Інтелектуальний агент надає також візуалізацію прогалин у знаннях про всі нефункційні характеристики. Така візуалізація є аналогічною до візуалізації для першого експерименту, представленої на рис. 2–9. Така візуалізація надає користувачу список відсутніх у специфікації вимог атрибутів для визначення нефункційних характеристик-складових якості ПЗ. Крім цього, саме така візуалізація відображає на скільки та які підхарактеристики впливає той чи інший атрибут. Після аналізу наданої агентом візуалізації прогалин у знаннях розробники можуть визначити, які атрибути першочергово слід внести до специфікації вимог з метою збільшення рівня достатності інформації.

6. Обговорення результатів функціонування розробленого інтелектуального агента на основі онтологічного підходу

Розроблена модель діяльності та реалізований інтелектуальний агент на основі онтологічного підходу забезпечили можливість автоматизувати аналіз специфікацій вимог до ПЗ на предмет достатності їх інформації. Автоматизація аналізу специфікацій вимог до ПЗ стала можливою за рахунок використання онтологій при функціонуванні розробленого агента. Саме онтології, за рахунок відображення причинно-наслідкових зв'язків між поняттями та концептуалізації домена фіксуванням сутностей та зв'язків, забезпечили виявлення дублювань та прогалин у знаннях на основі візуалізації відсутніх логічних зв'язків. Така візуалізація відсутніх логічних зв'язків відображає, яких саме атрибутів недостатньо у специфікації, на які нефункційні характеристики-складові якості ПЗ впливає відсутність певних атрибутів та який рівень достатності інформації у певній специфікації.

Реалізований інтелектуальний агент спрямований саме на автоматизацію оцінювання достатності інформації щодо нефункційних характеристик-складових якості ПЗ у специфікаціях вимог, на відміну від агентів, розроблених у [15–20]. Таким чином, він є розвитком концепції оцінювання достатності інформації, представленої у [5, 13].

Інтелектуальність реалізованого інтелектуального агента полягає не тільки у формуванні висновку про достатність або недостатність інформації та наданні числових оцінок рівня достатності інформації у специфікації вимог. Він також рекомендує доповнити цю специфікацію атрибутами, необхідними для визначення нефункційних характеристик-складових якості ПЗ (з наданням списку та візуалізації відсутніх атрибутів), як показав проведений у розділі 5 аналіз. Якщо розробники специфікації дослухаються до рекомендацій агента і доповнюють специфікацію необхідними атрибутами, то рівень достатності інформації у аналізованій специфікації вимог для визначення нефункційних характеристик-складових якості ПЗ зростає.

Наприклад, нехай розробники додали у специфікацію вимог до ПЗ транспортно-логістичної системи ТОВ «Гілея», розглянуту вище, наступні атрибути для Security: Access Controllability, Number of Access Types, Number of Provided Authentication Methods, Number of Events Processed Using Digital Signature. Тоді числова оцінка рівня достатності інформації у цій специфікації після доповнення становитиме для визначення Security:

$$D_{Scr} = \frac{\left(5 - \left(\frac{2}{10} + \frac{2}{8} + \frac{0}{2} + \frac{0}{2} + \frac{0}{1}\right)\right)}{5} = 0,91.$$

Таким чином, доповнення специфікації вимог чотирма атрибутами для Захищеності (Security) підвищить рівень достатності інформації для визначення Захищеності (Security) на 39 % – з 52 % до 91 %. Відповідно підвищення рівня достатності інформації для однієї з нефункційних характеристик приводитиме

до підвищення рівня достатності інформації для визначення всіх нефункційних характеристик-складових якості ПЗ. Крім цього, підвищення рівня достатності інформації для однієї з нефункційних характеристик може позитивно впливати й на рівень достатності інформації для інших нефункційних характеристик. Адже використовувані онтології якраз і проявляють кореляцію нефункційних характеристик за атрибутами, тобто підказують користувачу, які атрибути потрібно внести у специфікацію першочергово для більш стрімкого збільшення достатності інформації. А підвищення рівня достатності інформації для декількох нефункційних характеристик приводитиме до ще більшого зростання рівня достатності інформації для визначення всіх нефункційних характеристик-складових якості ПЗ.

Отже, розроблений інтелектуальний агент на основі онтологічного підходу забезпечує підвищення рівня достатності інформації у специфікації вимог до ПЗ для визначення восьми нефункційних характеристик-складових якості ПЗ. В цьому полягає його відмінність від відомих підходів, описаних у [4, 21], які забезпечують лише оцінювання достатності інформації для однієї нефункційної характеристики ПЗ (безпеки), та описаного у [22], спрямованого на визначення достатності тестування з врахуванням верифікації ПЗ до поставлених вимог.

Розроблений інтелектуальний агент на основі онтологічного підходу може використовуватись для будь-якого програмного забезпечення, єдиним обмеженням є наявність специфікації вимог до ПЗ.

Недоліком представленого рішення є те, що на сьогодні вибір інформації щодо нефункційних характеристик зі специфікації вимог до реального ПЗ для формування реальних онтологій реалізується вручну. Для цього користувач відкриває базову онтологію предметної галузі «Інженерія програмного забезпечення» (частина «Якість ПЗ») та переглядає відповідний розділ специфікації на предмет наявності атрибутів, зазначених у базовій онтології. Надалі планується автоматизація цього етапу – буде розроблено інший агент, який проводитиме семантичний розбір природомовної специфікації на предмет пошуку атрибутів, необхідних для визначення нефункційних характеристик-складових якості ПЗ.

7. Висновки

Розроблено модель діяльності інтелектуального агента на основі онтологічного підходу для оцінювання специфікацій вимог до ПЗ, яка ґрунтується на порівняльному аналізі онтологій та є теоретичним підґрунтям для реалізації інтелектуального агента на основі онтологічного підходу. Реалізовано інтелектуальний агент, який працює на основі розробленої моделі та здійснює оцінювання достатності інформації у специфікації вимог для визначення всіх нефункційних характеристик-складових якості ПЗ. Реалізований інтелектуальний агент забезпечує висновок про достатність або недостатність інформації у специфікації. Крім цього, він надає числові оцінки рівня достатності інформації для визначення кожної нефункційної характеристики ПЗ та для визначення всіх нефункційних характеристик-складових якості ПЗ разом. Агентом також формується список атрибутів, якими

варто доповнити специфікацію вимог для підвищення рівня достатності її інформації, та візуалізація прогалін у знаннях про всі нефункційні характеристики-складові якості ПЗ. Отже, реалізований інтелектуальний агент забезпечує автоматизацію аналізу специфікацій вимог до ПЗ на предмет достатності їх інформації щодо нефункційних характеристик-складових якості ПЗ. Таким чином, представлений агент дозволяє частково усунути людину з процесів опрацювання інформації та здобуття знань.

Проаналізовано інформацію про нефункційні характеристики у специфікації вимог до ПЗ з використанням розробленого інтелектуального агента на основі онтологічного підходу. Проведений аналіз показав, що всі результати функціонування реалізованого інтелектуального агента на основі онтологічного підходу в комплексі забезпечують підвищення рівня достатності інформації у специфікації вимог до ПЗ для визначення нефункційних характеристик-складових якості ПЗ. Крім цього, результати функціонування реалізованого інтелектуального агента спрямовані на уникнення втрат істотної інформації і мінімізацію виникнення помилок на ранніх етапах життєвого циклу ПЗ.

Література

1. Hastie S., Wojewoda S. Standish Group 2015 Chaos Report – Q&A with Jennifer Lynch. URL: <http://www.infoq.com/articles/standish-chaos-2015>
2. McConnell S. Code complete. Redmond, 2013. 896 p.
3. Levenson N. G. Engineering a safer world: systems thinking applied to safety. Cambridge, 2012. 560 p.
4. Cruickshank K. J. A validation metrics framework for safety-critical software-intensive systems. Monterey, 2009. 144 p.
5. Hovorushchenko T., Pomorova O. Method of evaluating the success of software project implementation based on analysis of specification using neuronet information technologies // CEUR-WS. 2018. Vol. 2104. P. 555–570.
6. ISO/IEC 25010:2011. Systems and Software Engineering. Systems and Software Quality Requirements and Evaluation (SQuaRE). System and Software Quality Models. Geneva, 2011. 34 p.
7. Gruber T. R. A translation approach to portable ontology specifications // Knowledge Acquisition. 1993. Vol. 5, Issue 2. P. 199–220. doi: <https://doi.org/10.1006/knac.1993.1008>
8. Burov E. Complex ontology management using task models // International Journal of Knowledge-Based and Intelligent Engineering Systems. 2014. Vol. 18, Issue 2. P. 111–120. doi: <https://doi.org/10.3233/KES-140291>
9. Burov E., Pasitchnyk V., Gritsyk V. Modeling software testing processes with task ontologies // British Journal of Education and Science. 2014. Vol. 2, Issue 6. P. 256–263.
10. Assawamekin N., Sunetnanta T., Pluempitiwiriyawej C. Ontology-based multiperspective requirements traceability framework // Knowledge and Information Systems. 2009. Vol. 25, Issue 3. P. 493–522. doi: <https://doi.org/10.1007/s10115-009-0259-2>

11. Ontology and Model Alignment as a Means for Requirements Validation / Kof L., Gacitua R., Rouncefield M., Sawyer P. // 2010 IEEE Fourth International Conference on Semantic Computing. 2010. doi: <https://doi.org/10.1109/icsc.2010.95>
12. An ontological approach to model software quality assurance knowledge domain / Bajnaid N. O., Benlamri R., Pakstas A., Salekzamankhani Sh. // Lecture Notes on Software Engineering. 2016. Vol. 4, Issue 3. P. 193–198.
13. Hovorushchenko T., Pomorova O. Ontological approach to the assessment of information sufficiency for software quality determination // CEUR-WS. 2016. Vol. 1614. P. 332–348.
14. Wooldridge M., Jennings N. R. Intelligent agents: theory and practice // The Knowledge Engineering Review. 1995. Vol. 10, Issue 2. P. 115–152. doi: <https://doi.org/10.1017/s0269888900008122>
15. Freitas A., Bordini R. H., Vieira R. Model-driven engineering of multi-agent systems based on ontologies // Applied Ontology. 2017. Vol. 12, Issue 2. P. 157–188. doi: <https://doi.org/10.3233/ao-170182>
16. Exploring an Ontological Approach for User Requirements Elicitation in the Design of Online Virtual Agents / Ossowska K., Szewc L., Weichbroth P., Garnik I., Sikorski M. // Lecture Notes in Business Information Processing. 2016. P. 40–55. doi: https://doi.org/10.1007/978-3-319-46642-2_3
17. Lezcano-Rodriguez L. A., Guzman-Luna J. A. Ontological characterization of basics of KAOS chart from natural language // ITECKNE. 2016. Vol. 13, Issue 2. P. 157–168. doi: <https://doi.org/10.15332/iteckne.v13i2.1482>
18. García-Magariño I., Gómez-Sanz J. J. An Ontological and Agent-Oriented Modeling Approach for the Specification of Intelligent Ambient Assisted Living Systems for Parkinson Patients // Lecture Notes in Computer Science. 2013. P. 11–20. doi: https://doi.org/10.1007/978-3-642-40846-5_2
19. Rakib A., Faruqui R. U. A Formal Approach to Modelling and Verifying Resource-Bounded Context-Aware Agents // Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. 2013. P. 86–96. doi: https://doi.org/10.1007/978-3-642-36642-0_9
20. A Task-based Support Architecture for Developing Point-of-care Clinical Decision Support Systems for the Emergency Department / Michalowski W., O’Sullivan D., Farion K., Sayyad-Shirabad J., Kuziemy C., Kukawka B., Wilk S. // Methods of Information in Medicine. 2013. Vol. 52, Issue 01. P. 18–32. doi: <https://doi.org/10.3414/me11-01-0099>
21. Hazard Analysis and Validation Metrics Framework for System of Systems Software Safety / Michael J. B., Shing M.-T., Cruickshank K. J., Redmond P. J. // IEEE Systems Journal. 2010. Vol. 4, Issue 2. P. 186–197. doi: <https://doi.org/10.1109/jsyst.2010.2050159>
22. Baker R., Habli I. An Empirical Evaluation of Mutation Testing for Improving the Test Quality of Safety-Critical Software // IEEE Transactions on Software Engineering. 2013. Vol. 39, Issue 6. P. 787–805. doi: <https://doi.org/10.1109/tse.2012.56>

23. ISO 25023:2016. Systems and Software Engineering. Systems and Software Quality Requirements and Evaluation (SQuaRE). Measurement of System and Software Product Quality. Geneva, 2016. 45 p.

24. Hovorushchenko T., Pavlova O. Method of Activity of Ontology-Based Intelligent Agent for Evaluating Initial Stages of the Software Lifecycle // Advances in Intelligent Systems and Computing. 2019. P. 169–178. doi: https://doi.org/10.1007/978-3-319-97885-7_17

25. ISO/IEC/IEEE 29148:2011. Systems and Software Engineering. Life Cycle Processes. Requirements Engineering. Geneva, 2011. 28 p.

Т О Л Ь К О Д Л Я Ч Т Е Н И Я