

## СИСТЕМА ПІДТРИМКИ СИТУАЦІЙНОЇ ОБІЗНАНОСТІ У ПРОЦЕСІ ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Євген Буров<sup>1</sup>, Христина Микіч<sup>2</sup>, Олег Верес<sup>3</sup>

<sup>1,2,3</sup> Національний університет “Львівська політехніка”

<sup>1</sup> Yevhen.V. Burov@lpnu.ua, ORCID 0000-0001-8653-1520

<sup>2</sup> Khrystyna.I. Mykich@lpnu.ua, ORCID 0000-0002-4324-2080

<sup>3</sup> Oleh.M. Veres@lpnu.ua, ORCID 0000-0001-9149-4752

© Буров Є., Микіч Х., Верес О., 2020

Досліджено та розроблено прототип інформаційної системи, що базується на онтологічному моделюванні із використанням логічного виведення (deskриптивної логіки) у процесі тестування програмного забезпечення.

Розглянуто актуальну проблему використання ситуаційної обізнаності як ключового фактору при проектуванні системи підтримки прийняття рішення.

Для практичного застосування розроблених методів ідентифікації критичних ситуацій обрано галузь тестування програмного забезпечення, що пов'язано із складністю процесів під час розроблення програмного забезпечення та високою ціною помилки. Системи тестування програмного забезпечення відіграють центральну роль під час його розроблення, адже використовуються для можливості виправляти помилки на ранньому етапі та впроваджувати нові функції, а також для контролю якості та управління проектами, для відстеження історії помилок. Якісний результат проектування базується на високих вимогах, а не тільки на навичках та знаннях розробника. Для прийняття високоякісних та правильних рішень програміст має мати високий рівень ситуаційної обізнаності.

Проаналізовано найвідоміші сучасні методи використання ситуаційної обізнаності, щоб пояснити ключові моменти в усвідомленні ситуації та правильності подання самої ситуації та прийняття рішення.

У результаті аналізу було виділено основні критерії, за якими і здійснювалося порівняння. Результати дослідження було зведено у порівняльну таблицю, що дасть змогу чітко визначити, який із методів потрібно використовувати, враховуючи цілі проектування програмного забезпечення. Недоліком проаналізованих методів є те, що вони не дають змоги використовувати в межах поточної системи різні типи ситуацій, що виникають у сучасному середовищі. Сьогодні за стрімкого розвитку інформаційних технологій та великої кількості даних це вкрай необхідно. Саме тому розроблена у роботі система була націлена на розв'язання цієї наукової задачі.

Розроблений прототип інформаційної системи дасть змогу розробникам програмного забезпечення співпрацювати, одночасно покращуючи загальну обізнаність про поточний стан системи та взаємодію між розвитком та операціями протягом усього процесу розробки.

Подані у роботі методи, на яких базується розроблений прототип системи, дають змогу зберігати та використовувати знання про предметну область галузі тестування програмного забезпечення, а також дають можливість використовувати різні типи ситуацій у цілісній формі з врахуванням взаємозалежностей між об'єктами та ситуаціями, поданими у формі відношень. Крім того, застосування онтологій для ідентифікації ситуацій надає додаткові можливості для задання та опрацювання інформації про ситуації шляхом використання структурних особливостей та механізмів логічного виведення онтології.

**Результати роботи доцільно використовувати для розв'язання задач виявлення критичних ситуацій під тестування програмного забезпечення, що дасть змогу зменшити похибку ідентифікації порівняно із традиційними методами ідентифікації.**

**Ключові слова:** ситуаційна обізнаність, тестування програмного забезпечення, онтологія, логічне міркування, дескриптивна логіка.

### Вступ

Сьогодні ситуаційна обізнаність є основною вимогою щодо прийняття рішень у відповідному просторі часу. Ситуаційну обізнаність використовують для створення надійної картини ситуації та ухвалення правильних рішень з метою уникнення небезпеки, що відповідатимуть найвищим стандартам якості.

Ситуаційна обізнаність є достатньо новим напрямком дослідження. Всі дослідження, що проводилися дотепер, ґрунтувались на людино-машинній взаємодії, внаслідок чого виникла необхідність у розробленні таких систем підтримки прийняття рішень, які є повністю автоматизованими та мають змогу використовувати попередній досвід з ідентифікації критичних ситуацій.

На рис. 1 показано динаміку зростання потреб у системах зі ситуаційною обізнаністю (відповідно до даних маркетингового дослідження компанії MarketsandMarkets) [1].

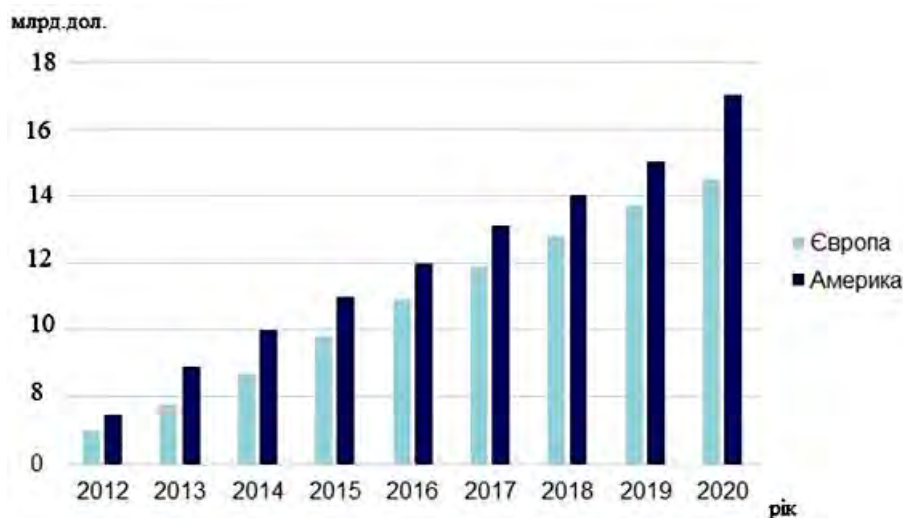


Рис. 1. Динаміка зростання потреб у системах із ситуаційною обізнаністю

Сьогодні зростає ринок систем зі використанням ситуаційної обізнаності (рис. 1). Зростання у всьому світі пов'язане із потребою у громадській безпеці та потребою розроблення нової програмної технології для збирання, аналізу, візуалізації, зберігання даних та оперативного управління, а також розробленням програмного забезпечення щодо керування критичними ситуаціями та подіями.

Отже, розроблення систем із використанням ситуаційної обізнаності та їх вдосконалення є актуальним, оскільки існує потреба в проектуванні таких систем.

### Постановка проблеми

Ситуаційна обізнаність – це сприйняття навколишнім середовищем подій стосовно простору чи часу. Це дослідження здійснюється для розуміння середовища, яке є критичним, та ухвалення рішення у складних ситуаціях, таких як управління повітряним рухом, експлуатація електростанцій, розроблення програмного забезпечення тощо. Усвідомлення поточної ситуації передбачає розуміння того, які події, інформація чи навіть власні дії впливають на цілі та завдання на поточному етапі чи в майбутньому. Особа, яка володіє досвідом обізнаності щодо ситуації, зазвичай має високий ступінь знань разом із належним розумінням того, що система отримує на вході та що потрібно отримати на виході. Відсутність ситуаційної обізнаності є основним фактором нещасних випадків через помилки людини.

Однією з невирішених задач є визначення подання ситуації в межах системи моделювання. Дуже часто системи моделювання орієнтовані на опис одного типу ситуацій, тоді як в реальних умовах існують різні типи ситуацій. Прийняття рішення та ідентифікація ситуації в реальних умовах передбачають поєднання різних методів.

Отже, ситуаційна обізнаність відіграє життєво важливу роль у сферах інформаційних технологій, оскільки сьогодні потік інформації є доволі високий, а неправильні рішення можуть призвести до серйозних наслідків.

### **Аналіз останніх досліджень та публікацій**

Поінформованість щодо ситуації є найважливішою складовою у найрізноманітніших сферах, пов'язаних зі працездатністю особистості та команди, і ключовим фактором при розробленні системи та оцінюванні систем керування, навчальних програм тощо [2].

Рівень усвідомлення ситуації, який системи надають користувачам, є важливим фактором, що потребує оцінювання протягом усього процесу проектування. Існують різні методи оцінювання ситуаційної обізнаності, але мало вказівок щодо того, який із цих методів є найбільш достовірним, надійним та точним.

Вважають, що при виборі відповідного методу із використанням ситуаційної обізнаності потрібно враховувати характеристики завдання, цілі аналізу та проектування системи.

Для практичного застосування розроблених методів ідентифікації проблемних ситуацій обрано галузь тестування програмного забезпечення. Однією з найважливіших проблем галузі програмного забезпечення є високий рівень складності програмних систем та високої вартості адміністрування:

1. Складна конфігурація процесів. Необхідність приймати рішення щоденно. Велика відповідальність та ціна помилки.

2. Працівник QA (тестування програмного забезпечення) працює в умовах стресу та розвиває “тунельне бачення” невеликого кола задач. Часто не бачить усієї картини і може помилитися.

У роботі проаналізовано наявні фреймворки зі використанням ситуаційної обізнаності. В результаті аналізу було визначено такі основні характеристики, за якими й здійснювалося порівняння, а саме:

- *повторне використання знань* – цю характеристику використовують для збереження інформації про ситуації, що відбувались у минулому, з метою врахування помилок і уникнення повторного їх використання в майбутньому;
- *врахування історії* – цю характеристику використовують для визначення історії станів у різні моменти часу системи, оскільки поведінка об'єкта – це складний процес, на який впливають різноманітні стани в різний момент часу;
- *логічне виведення* – цю характеристику використовують для логічного відслідковування різноманітних відношень між сутностями, що дає змогу вирішити завдання класифікації даних;
- *використання правил* – цю характеристику використовують для дослідження взаємозв'язку між подіями, що відбуваються із застосуванням логічних правил;
- *часово-просторові* – цю характеристику використовують для часово-просторових систем (рухомих систем), для яких, відповідно, змінюється стан із врахуванням простору та часу;
- *міркування на основі подібностей* – цю характеристику використовують для демонстрації ефективності розробленої системи, тобто за допомогою певної міри, зіставляють отримані результати із певними шаблонними показниками;
- *здатність навчатися* – характерна для системи, яка використовує різні алгоритми машинного навчання, що орієнтовані у цьому контексті на самостійне навчання. Тобто, як правило, ми маємо велику кількість даних і за певним методом “навчаємо” вибірку вхідних даних;
- *підтримка різних форм міркувань* – ця характеристика орієнтована на використання в межах єдиної системи різних форм міркувань, наприклад, логічних міркувань, причинно-

наслідкових міркувань, міркувань в умовах невизначеності тощо, що дає змогу працювати з різними типами даних;

- *спрощення процесу ідентифікації* – ця характеристика орієнтована на використання різних методів з метою спрощення процесу ідентифікації ситуації та менеджменту знань про такі ситуації;

- *нечіткість та неповнота даних* – у реальному світі дані, отримані із сенсорів, можуть бути нечіткими (неможливо точно віднести до певного класу) чи неповними (відсутні певні дані, що унеможлиблює точну їх класифікацію);

- *врахування контексту* – ця характеристика зосереджена на використанні контекстної інформації для того, щоб прийняти правильні рішення, оскільки під час розроблення автоматизованих систем часто виникає питання про те, як потрібно описати і відобразити певну ситуацію, щоб система могла її ідентифікувати і чи потрібно описувати тільки поточну ситуацію чи також розглядати її попередні стани, чи можна інформацією, яка є неважливою, знехтувати. Тобто, врахування контексту дає змогу адаптуватись системі до певних характеристик середовища та потрібної інформації.

Проведемо порівняльний аналіз деяких наявних фреймворків побудови інтелектуальних систем із ситуаційною обізнаністю.

1. Фреймворк із використанням часової логіки як методу дослідження запропонував автор Jaroucheh [3].

Основною ідеєю запропонованого фреймворку є те, що на основі інформації про стан відповідного процесу із застосуванням певного методу видобування знань (mining algorithms) отримується модель процесу (збираючи існуючі контекстні стани журналів). Точність сформованої моделі залежить від кількості розглянутих станів та обраного алгоритму видобування даних [3].

Цей фреймворк відповідає таким трьом вищепереліченим характеристикам:

- *врахування історії* – при побудові моделі процесу використовує так званий журнал (log), який містить стани поточної ситуації у різні моменти часу і дає змогу отримати загальну картину процесу та зрозуміти поведінку об'єкта;

- *врахування контексту* – врахування інформації, що орієнтована на стан середовища та нехтування тією інформацією, що, на думку розробника, є неважливою для ідентифікації ситуації;

- *врахування нечіткості та неповноти даних* – враховує те, що дані, отримані із середовища, можуть бути неповними та нечіткими і не всі стани ситуації, що ідентифікується, можна точно описати. Тобто, у цьому фреймворку ідентифікуються нечіткі ситуації – коли характеристики ситуації можуть не зовсім збігатися з шаблоном (є відхилення, які система вважає незначними).

2. Фреймворк із використанням онтології як методу дослідження, запропонований автором Baumgartner [4].

Основною ідеєю цього фреймворку є використання онтології базового рівня, в якій реалізовано базові мереотопологічні відношення, та зовнішніх баз знань, де зберігається інформація щодо предметної області для часово-просторових систем [4].

Цей фреймворк відповідає таким трьом характеристикам:

- *повторне використання знань* – використання онтології та зовнішніх баз знань, де зберігається інформація щодо предметної області та повторне її використання;

- *логічне виведення* – використання логічного виводу на базі правил із використанням мов OWL та DL;

- *часово-просторові* – використання просторово-часових базових зв'язків між спостережуваними реальними об'єктами;

- *спрощення процесу ідентифікації* – використання базових відношень як спрощення ідентифікації завдяки повторному використанню цих базових відношень у поданні складних ситуацій та відношень.

3. Фреймворк із використанням нейронних мереж як методу дослідження, запропонований автором Brannon [5].

Автор [5] використовує штучні нейронні мережі, що базуються на теорії адаптивного резонансу. Підхід, запропонований автором, полягає у використанні штучних нейронних мереж для навчання вибірки даних та використання відповідної наявної інформації без втручання людини-оператора.

Цей фреймворк відповідає тільки одній характеристиці:

- здатність навчатися – фреймворк складається з трьох рівнів, тобто поєднує в межах однієї системи три різні підходи машинного навчання та дає змогу опрацювати велику кількість даних, що здатні навчатися.

4. Фреймворк із використанням міркування на основі прецедентів (CBR – case-based reasoning) як методу дослідження, запропонований автором Nwiabu [6].

Основна ідея цього фреймворку полягає у використанні міркування на основі прецедентів, що дає змогу зрозуміти поточну ситуацію, порівнявши її з подібною ситуацією, яка відбулась у минулому [6].

Цей фреймворк відповідає таким характеристикам:

- повторне використання знань – наявна бібліотека ситуацій (Case-based situation library), яка містить ситуації, що відбувалися у минулому, та їх вирішення;
- врахування історії – для зберігання історії станів у цьому фреймворку існує певна компонента – стан середовища (State of the environment), яка накопичує сигнали про поточну ситуацію та надсилає її до іншої компоненти (Perception component);
- міркування на основі подібностей – використовується функція  $f(x)$ , яка знаходиться в межах  $[0; 1]$  і на її основі обчислюється загальна оцінка ситуації: найважливіші характеристики враховуються, а ті, які є менш важливі, опускаються;
- врахування контексту – дає змогу системі адаптуватись відповідно до конкретних потреб окремого оператора, оскільки ситуаційна обізнаність може мати різні значення для різних операторів у тому ж середовищі.

5. Фреймворк із використанням графу як методу дослідження, запропонований автором Tarapata [7].

Основна ідея фреймворку полягає у використанні багатокритеріальних зважених графів подібності та на основі шаблонів ситуацій прийняття рішень та поточної ситуації прийняття рішення шукається найбільш подібна шаблонна ситуація, яка міститься у базі даних [7].

Цей фреймворк відповідає таким характеристикам:

- повторне використання знань – повторне використання знань, поданих шаблонами;
- міркування на основі подібностей – у цьому фреймворку представлено підхід, що поєднує структурні та неструктурні подібності між графами. Подано міру обчислення подібності у зваженому графі.

6. Фреймворк із використанням асоціативних правил як методу дослідження, запропонований автором Feng [8].

Цей фреймворк складається із двох компонент: ситуаційної моделі та агентів сутностей. Ситуаційна модель базується на класичній моделі Ендслі, що складається з трьох рівнів [9], проте додатково ще містить модель місцевості (Terrain model). Агенти сутності, враховуючи контекст користувача, забезпечують його інформацією про поточну ситуацію [8].

Такий фреймворк відповідає таким характеристикам:

- використання правил – за допомогою виведення на основі правил, агенти сутностей забезпечують класифікацію даних та рекомендації щодо дій, пов'язаних із підтриманням прийняття рішень;
- врахування контексту – запропоновано загальну ситуаційну модель, яка забезпечує операторів інформацією за допомогою агентів сутностей, які існують для кожного конкретного оператора окремо. Агенти сутностей витягують інформацію, що є релевантною відповідно до контексту кожного користувача.

7. Фреймворк із використанням онтології як методу дослідження запропонований автором Joussemme [10].

Основна ідея фрейворку полягає у використанні онтології, яка базується на URREF (uncertainty representation and reasoning evaluation framework), що дає змогу зберігати інформацію про різні типи невизначеностей та їх оцінку [10].

Цей фреймворк відповідає таким характеристикам:

- повторне використання знань – використання онтології для представлення невизначеності, яка базується на URREF, та побудова додаткових елементів; дає змогу зберігати дані про невизначеність та повторно їх використовувати;
- нечіткість та неповнота даних – врахування різних типів невизначеності та їх представлення, а також міркування з використанням невизначеності за допомогою фреймворку URREF.

8. Фреймворк із використанням графу як методу дослідження, запропонований автором Nagi [11].

Основна ідея фреймворку полягає у тому, що дані подано за допомогою нечіткого графу. Нечіткі дані, отримані при спостереженні, злагоджуються та зіставляються з графовим шаблоном, вказуючи на невизначеність у ситуації [11].

Такий фреймворк відповідає тільки одній характеристиці:

- нечіткість та неповнота даних – використання нечітких графів для визначення невизначеності у поточній ситуації;

9. фреймворк із використанням онтології як методу дослідження, запропонований автором Dargie [12].

Основна ідея фреймворку полягає у використанні відомого методу дослідження – онтології. У роботі створено онтологію, що описує систему та сервери та використано дескриптивну логіку для ідентифікації ситуації. Використання дескриптивної логіки дає змогу працювати з неповними та нечіткими даними [12].

Цей фреймворк відповідає таким характеристикам:

- повторне використання знань – використання онтології для збереження інформації про систему та ситуації, що виникають, та можливість повторного їх використання;
- логічне виведення – використання дескриптивної логіки для ідентифікації ситуацій за допомогою мови опису онтологій OWL;
- нечіткість та неповнота даних – дає змогу враховувати нечіткість та неповноту даних за допомогою побудови аксіом, використовуючи мову опису онтологій OWL.

### **Формулювання цілі статті**

Метою роботи є розроблення прототипу інформаційної системи, що використовує онтологічне моделювання для формального подання знань про предметну область (тестування програмного забезпечення), використання даних знань у процесі ідентифікації та для використання логічного виведення.

### **Виклад основного матеріалу**

Онтологія предметної області використовується для збереження сутностей та відношень галузі тестування програмного забезпечення та використовується для побудови правил щодо виявлення ситуацій. Онтологія містить модель домену, представлену як таксономія класів [13]. Це створює можливість для однозначної інтерпретації всіх об'єктів предметної області з інформаційної бази, визначення для них загальних атрибутів та властивостей. Система реагує на певний діапазон подій (ситуацій, що виникають) у зовнішньому світі, ідентифікує їх та створює нові або змінює існуючі факти про неї (здійснює логічне міркування) [14].

Для створення онтології предметної області використано формальну мову OWL, яка є розширенням RDF/RDFS.

Розроблюваний фреймворк ґрунтується на онтології та алгебрі систем як методах дослідження.

Основна ідея розробленого фреймворку полягає у використанні онтології для подання предметної області та розробленні формальної моделі на базі алгебри систем, що дає змогу в межах єдиної системи поєднувати різні методи ідентифікації ситуацій [14].

Фреймворк відповідає таким характеристикам:

- повторне використання знань – використання онтології для подання предметної області та зберігання інформації про ситуації;
- врахування історії – використання підходу інтерпретованих систем для подання часової динаміки та розумування;
- логічне виведення – здійснення логічного міркування з використанням SWRL, базуючись на OWL;
- використання правил – побудова правил із використанням дескриптивної логіки;
- підтримка різних форм міркувань – розроблення формальної моделі подання знань про ситуації, яка дає змогу реалізувати моделювання, ідентифікацію та міркувати про ситуації для різних форм подання ситуацій та підтримує різні види міркувань;
- спрощення процесу ідентифікації – використання гранулярного комп'ютингу для задання ситуацій для великих груп об'єктів та апарату неточних множин у випадку, коли ознаки ситуації неточно задано;
- нечіткість та неповнота даних – врахування нечіткості та неповноти даних, моделювання процесів прийняття рішень у нечітких умовах.

Результати, отримані аналізом функціональних можливостей наявних фреймворків, наведено у таблиці.

### Порівняння функціональних можливостей фреймворків із використанням ситуаційної обізнаності

Автори	Повторне використання знань	Врахування історії	Логічне виведення	Використання правил	Часово-просторові	Міркування на основі подібностей	Здатність навчатися	Підтримка різних форм міркувань	Спрощення процесу ідентифікації	Нечіткість та неповнота даних	Врахування контексту
Jaroucheh (Часова логіка)		+								+	+
Vaumgarner (Онтології)	+		+		+				+		
Branon (Нейронні мережі)							+				
Nwiabu (Міркування на основі прецедентів)	+	+				+					+
Tararata (Граф)	+					+					
Feng (Асоціативні правила)				+							+
Jousselme (Онтології)	+									+	
Nagi (Граф)										+	
Waltenegus (Онтології)	+		+							+	
Розроблена система (онтології та алгебра систем)	+	+	+	+				+	+	+	

Запропонований підхід відповідає багатьом вимогам, які висувають до фреймворків побудови інтелектуальних систем зі ситуаційною обізнаністю, та основною його перевагою є гнучкість та поєднання різних методів у межах єдиної системи для з'ясування різних типів ситуацій, що виникають у середовищі.

Розроблена система виявлення ситуацій працює як консультант, який бачить загальну картину, та містить інформацію про попередні тестування. Вона виявляє типові паттерни, подані ситуаціями та повідомляє про це керівника відділу якості, топ-менеджера та розробників [14].

Вона також може використовуватися топ-менеджерами для виявлення катастрофічних ситуацій на проєкті, тому що часто керівник проєкту сподівається, що "якось буде" або просто не бачить загальної картини [14].

Користувачі системи: керівник відділу якості, топ-менеджер, керівник проєкту.

Вхідними даними системи є бази даних тестування та онтології.

Результат роботи: виявлення типових ситуацій та повідомлення користувачів.

Приклади ситуації на тестуванні (фрагмент), яка відповідає розробленим методам виявлення: недостатньо часу на проведення тестування перед релізом.

Дано:

- Дата релізу.
- Набір тесткейсів для перетестування.
- Кожен тесткейс має оцінку тривалості виконання.
- Кількість тестерів.
- Бюджет часу кожного тестера або призначені тесткейси.

Процедура: рахується час, необхідний для тестування, та порівнюється з тим, що залишилося.

Дія: повідомлення про ризик невиконання, визначення перевантажених тестерів.

П1. Множина протестованих тесткейсів:

$\text{TestCase}(?)^{\wedge}\text{notTested}(?, ?\text{NotTestedStatus}) \rightarrow \text{NotTested}(?)$

У базі знань містяться тесткейси, які вже були протестовані і які не є ще протестовані. В результаті виконання цього правила виведено ті тесткейси, які не є протестованими (рис. 2).

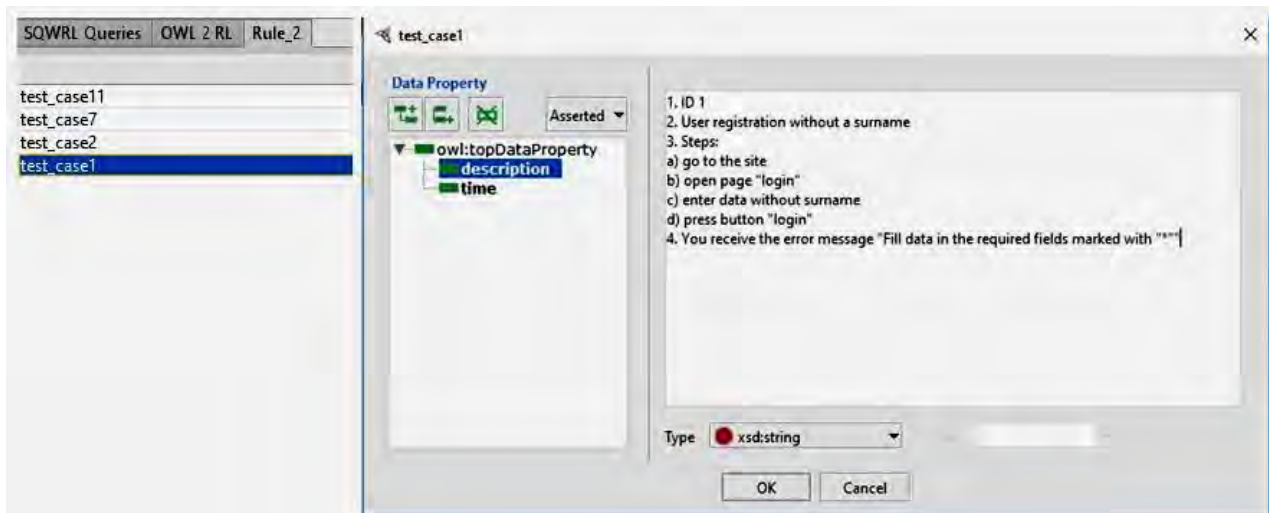


Рис. 2. Результат виконання програмної системи

Розроблений прототип інтелектуальної системи підтримки прийняття рішень дає змогу виявляти критичні ситуації на проєкті, а також повторно використовувати отримані знання з метою уникнення аналогічних ситуацій у майбутньому. Також система дає можливість ідентифікувати різні типи ситуацій, що виникають, а саме: нечіткі, неточні, залежні тощо.



### Висновки

Необхідність швидких та точних дій є невід'ємною частиною галузі інформаційних технологій на всіх організаційних рівнях. Тому сьогодні використання ситуаційної обізнаності при розробленні програмного забезпечення є ключовим фактором отримання якісного та надійного продукту. Компанії, що володіють розумінням поточної ситуації, мають чіткішу картину та кращий контроль над своїми системами, персоналом та результатами. Проаналізовані існуючі методи із використанням ситуаційної обізнаності показують, що цю галузь досліджено недостатньо. Розроблений прототип програмної системи є удосконаленим прототипом, оскільки дає змогу використовувати різні типи ситуацій на базі дескриптивної логіки та повторно використовувати знання.

### Список літератури

1. Situational Awareness Market (2015). Situational Awareness Market by Industry (Military & Defense, Aviation, Maritime Security, Automotive, Healthcare, Construction, Industrial, Homeland Security), by Components, Products, Applications, and Geography – Global Trend & Forecast to 2020. Retrieved from <https://www.marketsandmarkets.com/Market-Reports/situation-awareness-system-market-1272.html>
2. Верес, О. М. (2010). Види архітектури систем підтримки прийняття рішень. *Вісник Нац.ун-ту “Львівська політехніка”*. Комп'ютерні системи проектування. Теорія і практика. 685, 190–197.
3. Jaroucheh, Z., Liu, X., & Smith, S. (2011). Recognize contextual situation in pervasive environments using process mining techniques. *Journal of Ambient Intelligence and Humanized Computing*, 2(1), 53–69.
4. Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., & Schwinger, W. (2010). BeAware!—situation awareness, the ontology-driven way. *Data & Knowledge Engineering*, 69(11), 1181–1193.
5. Brannon, N. G., Seiffert, J. E., Draelos, T. J., & Wunsch II, D. C. (2009). Coordinated machine learning and decision support for situation awareness. *Neural Networks*, 22(3), 316–325.
6. Nwiabu, N., Allison, I., Holt, P., Lowit, P., & Oyeneyin, B. (2012, March). Case-based situation awareness. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support* (pp. 22–29). IEEE.
7. Tarapata, Z. (2007, August). Multicriteria weighted graphs similarity and its application for decision situation pattern matching problem. In *Proceedings of the 13th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics* (pp. 1149–1155).
8. Feng, Y. H., Teng, T. H., & Tan, A. H. (2009). Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, 36(1), 455–463.
9. Endsley, M. R., & Garland, D. J. (2000). Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, 1, 24.
10. Jousselme, A. L. (2016, July). Semantic criteria for the assessment of uncertainty handling fusion models. In *2016 19th International Conference on Information Fusion (FUSION)* (pp. 488–495). IEEE.
11. Nagi, R., Gross, G., & Sambhoos, K. (2014). A fuzzy graph matching approach in intelligence analysis and maintenance of continuous situational awareness. *Information Fusion*, 18, 43–61.
12. Dargie, W., Mendez, J., Möbius, C., Rybina, K., Thost, V., & Turhan, A. Y. (2013, March). Situation recognition for service management systems using OWL 2 reasoners. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (pp. 31–36). IEEE.
13. Lytvyn, V., Vysotska, V., & Veres, O. (2018). Ontology of big data analytics. *MEST Journal*, 6 (1), 41–60.
14. Буров, С. В., Микіч, Х. І., Верес, О. М., & Литвин, В. В. (2019). Система ідентифікації проблемних ситуацій тестування програмного забезпечення. *Вісник НУ “Львівська політехніка”*. Серія : “Інформаційні системи та мережі”, 5, 78–89. <https://doi.org/10.23939/sisn2019.02.030>

### References

1. Situational Awareness Market (2015). Situational Awareness Market by Industry (Military & Defense, Aviation, Maritime Security, Automotive, Healthcare, Construction, Industrial, Homeland Security), by Components, Products, Applications, and Geography – Global Trend & Forecast to 2020. Retrieved from <https://www.marketsandmarkets.com/Market-Reports/situation-awareness-system-market-1272.html>
2. Veres, O. (2010). Types of architecture for decision support systems. Bulletin of the National University “Lviv Polytechnic”. Series: “Computer-aided design systems. Theory and Practice”, 685, 190–197.
3. Jaroucheh, Z., Liu, X., & Smith, S. (2011). Recognize contextual situation in pervasive environments using process mining techniques. *Journal of Ambient Intelligence and Humanized Computing*, 2(1), 53–69.

4. Baumgartner, N., Gottesheim, W., Mitsch, S., Retschitzegger, W., & Schwinger, W. (2010). BeAware!—situation awareness, the ontology-driven way. *Data & Knowledge Engineering*, 69(11), 1181–1193.
5. Brannon, N. G., Seiffert, J. E., Draelos, T. J., & Wunsch II, D. C. (2009). Coordinated machine learning and decision support for situation awareness. *Neural Networks*, 22(3), 316–325.
6. Nwiabu, N., Allison, I., Holt, P., Lowit, P., & Oyenyin, B. (2012, March). Case-based situation awareness. In *2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support* (pp. 22–29). IEEE.
7. Tarapata, Z. (2007, August). Multicriteria weighted graphs similarity and its application for decision situation pattern matching problem. In *Proceedings of the 13th IEEE/IFAC International Conference on Methods and Models in Automation and Robotics* (pp. 1149–1155).
8. Feng, Y. H., Teng, T. H., & Tan, A. H. (2009). Modelling situation awareness for Context-aware Decision Support. *Expert Systems with Applications*, 36(1), 455–463.
9. Endsley, M. R., & Garland, D. J. (2000). Theoretical underpinnings of situation awareness: A critical review. *Situation awareness analysis and measurement*, 1, 24.
10. Joussetme, A. L. (2016, July). Semantic criteria for the assessment of uncertainty handling fusion models. In *2016 19th International Conference on Information Fusion (FUSION)* (pp. 488–495). IEEE.
11. Gross, G., Nagi, R., & Sambhoos, K. (2014). A fuzzy graph matching approach in intelligence analysis and maintenance of continuous situational awareness. *Information Fusion*, 18, 43–61.
12. Dargie, W., Mendez, J., Möbius, C., Rybina, K., Thost, V., & Turhan, A. Y. (2013, March). Situation recognition for service management systems using OWL 2 reasoners. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* (pp. 31–36). IEEE.
13. Lytvyn, V., Vysotska, V., & Veres, O. (2018). Ontology of big data analytics. *MEST Journal*, 6 (1), 41–60.
14. Burov, Ev., Mykich, Kh., Veres, O., & Lytvyn V. (2019). Situation Identification System in the Software Testing. *Bulletin of the National University “Lviv Polytechnic”. Series: “Information Systems and Networks”*, 5, 78–89. <https://doi.org/10.23939/sisn2019.02.030>

## SITUATION AWARENESS SUPPORT SYSTEM IN THE SOFTWARE TESTING PROCESS

Yevhen Burov<sup>1</sup>, Khrystyna Mykich<sup>2</sup>, Oleh Veres<sup>3</sup>

<sup>1,2,3</sup> Lviv Polytechnic National University

<sup>1</sup> Yevhen.V. Burov@lpnu.ua, ORCID 0000-0001-8653-1520

<sup>2</sup> Khrystyna.I. Mykich@lpnu.ua, ORCID 0000-0002-4324-2080

<sup>3</sup> Oleh.M. Veres@lpnu.ua, ORCID 0000-0001-9149-4752

© Burov Y., Mykich K., Veres O., 2020

**The paper is devoted to the development of a prototype information system based on ontological modeling using logical inference (descriptive logic) in the process of software testing.**

**The actual problem of using situational awareness as a key factor in designing the decision support system is considered.**

**For the practical application of the developed methods of critical situations identification the software testing industry has been selected. It is related to the complexity of the software development processes and the high cost of error.**

**Software testing systems play a central role in development, as they are used for the ability to correct errors on the early stage and introduce new features. As well as it used for quality control, project management, and tracking error history.**

**The quality design result is based on high requirements, not only on the skills and knowledge of the developer. To make high quality and correct decisions, the programmer requires a high level of situational awareness.**

**The paper analyzes the most well-known modern methods of using situational awareness to explain the key points in the situation awareness, the correct presentation of the situation itself and the correctness of decision making.**

As a result of the analysis, the most important criteria were identified and compared. The research was summarized in a comparative table to identify which methods to use, taking into account the goals of software design. The disadvantage of the analyzed methods is that they don't allow the use of different types of situations that arise in the current environment within the current system. Currently, with the rapid development of information technology and large amounts of data, this is essential. For this reason, the system developed in the paper was aimed at solving this scientific problem.

The developed information system prototype will enable software developers to collaborate while improving overall awareness of the current state of the system and interacting throughout the development process.

The methods discussed in the paper on which based the developed prototype system allow to store and use knowledge of the subject area of software testing.

They allow to use different types of situations in a holistic form, taking into account the interdependencies between objects and situations presented in the form of relations. Besides, the usage of ontologies to identify situations provides additional opportunities for specifying and processing information about situations by applying the structural features and mechanisms of logical inference to ontology.

It is advisable to use the results of the paper to solve the problems of identifying critical situations in software testing, which will help to reduce the errors of identification when compared with the traditional methods of identification.

**Key words:** situation awareness, software testing, ontology, logical inference, descriptive logic.