

4. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ



Науковий вісник НЛТУ України
Scientific Bulletin of UNFU

<http://nv.nltu.edu.ua>

<https://doi.org/10.15421/40290729>

Article received 14.08.2019 p.

Article accepted 26.09.2019 p.

УДК 004.[021+942]



ISSN 1994-7836 (print)
ISSN 2519-2477 (online)

@ ✉ Correspondence author

V. M. Teslyuk

vasyl.teslyuk@gmail.com

В. М. Теслюк, І. Г. Цмоць, А. Г. Казарян, Т. В. Теслюк

Національний університет "Львівська політехніка", м. Львів, Україна.

МЕТОД ПРОЕКТУВАННЯ СИСТЕМ "РОЗУМНОГО" БУДИНКУ З ВИКОРИСТАННЯМ АРХІТЕКТУРНОГО ШАБЛОНУ REDUX

Розроблено метод проектування систем "розумного" будинку з використанням архітектурного шаблону Redux. Метод ґрунтується на адаптації архітектурного шаблону Redux, що застосовується для проектування візуальних інтерфейсів до використання у сфері Інтернету речей. На підставі розробленого методу побудовано систему "розумного" будинку для управління освітлювальними приладами за допомогою давачів руху та освітлення у приміщеннях офісної будівлі. Розроблений метод проектування дає змогу підвищити показники надійності та швидкодії роботи системи. Покращення надійності досягається завдяки зниженню кількості прямих взаємозв'язків між компонентами системи. Також розроблений метод проектування сприяє зниженню обсягу інформації, яка дублюється у різних компонентах проектованої системи, завдяки використанню одного загального сховища даних для збереження стану, за рахунок чого підвищується швидкість оновлення станів системи та швидкість зміни налаштувань освітлювальних приладів. Переваги використання методу проектування експериментально відображено за допомогою емуляції роботи системи, "розумного" будинку з подальшим збереженням та аналізом показників швидкості зміни налаштувань освітлювальних приладів – до застосування та після застосування архітектурного шаблону Redux. Запропонований метод дає змогу масштабувати систему додаючи нові давачі та побутові прилади до розробленої системи без втрати швидкості опрацювання даних та передачі команд керування приладами. Розглянутий у роботі приклад надає перевагу запровадженню методу для проектування систем "розумного" будинку, що застосовуватимуться у сфері масового обслуговування, надаючи функціональність автоматизованого керування приладами у великих житлових, адміністративних і офісних будівлях з загальною характерною рисою великої кількості одночасно виникаючих подій, які надходять до системи для подальшого опрацювання та потребують відповідних змін станів налаштувань системи "розумного" будинку.

Ключові слова: проектування; архітектурний шаблон; Redux; "розумний" будинок.

Вступ. Ми живемо в час невідомого розвитку інтелектуальних технологій (Poniszewska-Maranda et al., 2018; Boreiko et al., 2017; Cai et al., 2017). Інтелектуалізація дає змогу істотно покращити функціональність систем, зменшити витрати споживаної енергії (Teslyuk et al., 2018) тощо. Однією з таких технологій є системи "розумного" будинку (СРБ) (Sultan & Ahmed, 2017; Kazarian et al., 2017). Вимоги до систем "розумного" будинку стають дедалі жорсткішими, тоді як розробники систем цього типу змушені керувати щоразу більшою кількістю станів системи в окремі моменти часу її роботи (Teslyuk et al., 2010). Ці стани можуть охоплювати дані, отримані від здавачів, розташованих всередині будинку, збережені, так звані, історичні дані показників давачів, а також інформацію, генеровану в процесі ро-

боти системи, таку як команди зміни режимів роботи побутових приладів у будинку.

Написання логіки керування станами системи, що часто змінюються, – це складна частина процесу розроблення СРБ, яка потребує значних часових та фінансових витрат. Значення стану давача потребує оновлення режиму роботи побутового приладу в будинку, що може змінити значення стану іншого давача, який вимагає оновлення режиму роботи іншого побутового приладу в будинку. На певному етапі розроблення програмного забезпечення "розумного" будинку, проєктувальники не знають, що відбувається всередині логіки роботи системи та не можуть контролювати коли, чому і як стани системи оновлюються. Така ситуація, коли робота системи стає непрозорою, зазнає величезних ри-

Інформація про авторів:

Теслюк Василь Миколайович, д-р техн. наук, професор, кафедра систем автоматизованого проектування.

Email: vasyli.m.teslyuk@lpnu.ua; <https://orcid.org/0000-0002-5974-9310>

Цмоць Іван Григорович, д-р техн. наук, професор, завідувач кафедри автоматизованих систем управління.

Email: ivan.tsmots@gmail.com

Казарян Артем Геннадійович, аспірант, кафедра систем автоматизованого проектування. Email: ivan.tsmots@gmail.com

Теслюк Тарас Васильович, аспірант, кафедра автоматизованих систем управління. Email: taras.teslyuk@gmail.com;

<https://orcid.org/0000-0001-6585-3715>

Цитування за ДСТУ: Теслюк В. М., Цмоць І. Г., Казарян А. Г., Теслюк Т. В. Метод проектування систем "розумного" будинку з використанням архітектурного шаблону Redux. Науковий вісник НЛТУ України. 2019, т. 29, № 7. С. 146–150.

Citation APA: Teslyuk, V. M., Tsmots, I. H., Kazarian, A. G., & Teslyuk, T. V. (2019). The Method of Smart Home Systems Design Using the Redux Architectural Pattern. *Scientific Bulletin of UNFU*, 29(7), 146–150. <https://doi.org/10.15421/40290729>

зиків щодо надійності роботи самої системи (Event-Based Systems, 2019), швидкості виявлення і виправлення помилок розробленої логіки та можливості розробляти нову функціональність. Ця складність виникає через те, що змішуються дві різні за сутністю концепції: зміни й асинхронності. Під змінами вважають потік генерованих у режимі реального часу показників даних від давачів, що розташовані в будинку, які впливають на зміни налаштувань побутових приладів у будинку. Під асинхронністю вважають неперіодичність виникнення цих змін показників, тобто зміна параметра давача відбувається за фактом виникнення подій у будинку (рух людини, зміна температури в кімнаті), що не залежать від часових інтервалів та не мають чіткого часового розкладу. Для вирішення цієї проблеми пропонують використати архітектурний шаблон Redux з адаптацією його концепцій до сфери розроблення систем "розумного" будинку (Islam Naim, 2017; Piispanen, 2017).

Проблеми швидкодії роботи систем, які оперують багатьма подіями, що одночасно виникли, описано в роботах (Paul & Nalwaya, 2016; Freeman, 2019; Saransig & Tapia, 2019), що пропонують приклади та порівняння використання монолітної, мікросервісної та багаторівневої архітектур. У сфері розроблення візуальних інтерфейсів користувача, проблема опрацювання великих масивів подій вирішується за допомогою використання Flux-подібних архітектур, таких як Redux, запровадження яких також розглянуто в роботах (Familiar, 2015; Nene, Joseph & Chandrasekaran, 2019). Проте роз-

глянуті концепції та приклади обмежені використанням тільки для проектування візуальних інтерфейсів.

Метод вибору та застосування архітектурного шаблону Redux до проектування систем "розумного" будинку. Для розроблення високонавантажених та масштабованих систем, до яких належать СРБ, використовують різні архітектурні рішення, які спершу застосовували у системах з інших сфер інформаційних технологій (Christudas, 2019; Kalske, Mäkitalo & Mikko-nen, 2018; Gackenheimer, 2015).

Кожен архітектурний підхід має недоліки та переваги, залежно від специфіки використання інформаційної системи, які різняться вимогами до показників швидкодії, надійності та масштабованості. Системи "розумного" будинку характеризуються багатьма одночасно активними подіями, що потребують опрацювання. Для вибору оптимального рішення щодо проектування, запропоновано впровадження кожного архітектурного підходу в СРБ, що працює на підставі однакової вибірки тестових даних, збирання та збереження параметрів швидкодії роботи системи під час використання кожного розглянутого рішення з подальшим аналізом отриманих результатів та вибором оптимального рішення щодо проектування. Вибір оптимального архітектурного рішення є кінцевою стадією процесу перебору декількох архітектурних рішень, з порівнянням параметрів швидкодії роботи системи. Розроблений алгоритм методу проектування архітектурного рішення зображено на рис. 1.



Рис. 1. Розроблений алгоритм вибору архітектурного рішення у процесі синтезу системи

з отриманих результатів аналізу видно, що найшвидші показники швидкодії роботи системи отримали під час запровадження у системі – Flux архітектури, а саме архітектурного шаблону Redux (Molnár et al., 2014). Найповільніше система працювала під час використання монолітної архітектури.

Отже, розглянутий підхід із використанням архітектурного шаблону потоку даних Redux робить потік даних зрозумілим для подальшого опрацювання відповідно до реалізованої логіки програмного забезпечення СРБ.

Адаптація концепцій архітектурного шаблону Redux відповідно до сфери систем "розумного" будинку. До базових концепцій архітектурного шаблону Redux відносять події, сховище та редюсер. Події – це структури, які передають дані у сховище. Вони є єдиними джерелами інформації для сховища. У сфері розроблення "розумних" будинків, під подією будемо розумі-

ти структуру даних, яка створюється генератором подій внаслідок виникнення ситуацій, таких як спрацювання давача чи команди користувача системи.

Сховище – це об'єкт, який зберігає стан системи, надає доступ до стану системи, має можливість оновлювати стан системи та дає змогу рееструватись у ролі "слухача" оновлення стану системи, щоб отримувати нові параметри стану системи після проведених змін. Redux використовує тільки одне сховище для всього стану параметрів системи.

Redux не дає змоги компонентам системи змінювати стан безпосередньо. Події описують те, які зміни необхідно зробити у параметрах стану системи. Обробників, які реагують на події і можуть вносити зміни у сховище, називають редюсерами. Вхідними параметрами редюсер обов'язково має приймати поточний стан системи, а також події. Відповідно до типу виникненої події, що має унікальну назву, редюсер виконує логічні дії

над параметрами отриманого стану та повертає на вихід змінений стан системи. Отримуючи аргументи визначеного типу, редюсер повинен обчислювати нову версію стану і передавати її сховищу даних. Жодних змін поточного стану системи (СС) редюсер виконувати не має, що забезпечує чітке розмежування функцій редюсера та сховища даних (СД). Робота редюсера обмежена тільки обчисленням нової версії стану, що перезаписує поточний стан системи у сховищі. Схему життєвого циклу СС відображено на рис. 2.

Отримуючи на вхід активовану подію, редюсер по чергово порівнює тип отриманої події з відомими редюсеру. У разі збігання типів, редюсер робить запит поточного СС та, використовуючи логіку зміни стану системи, що відповідає конкретному типу події, проводить зміни параметрів стану. Вихідним результатом є новий стан системи, що зберігається у СД, замінюючи попередній. Приклад алгоритму роботи редюсера, логіка якого відповідає за підтримку температурного режиму відповідно до отриманих даних від сенсора температури та за допомогою зміни налаштувань кондиціонера та обігрівача, наведено на рис. 3.

Отже, розроблена система "розумного" будинку з адаптацією концепцій архітектурного шаблону Redux

дає змогу ефективно керувати температурним режимом у приміщеннях будинку та дає змогу легко масштабувати систему для використання у будинку з більшою кількістю приміщень, без втрат швидкодії роботи завдяки запровадженню єдиного СС системи та стандартизованого підходу до його зміни незалежно від джерела подій, що виникають у будинку.



Рис. 2. Схема життєвого циклу стану системи "розумного" будинку

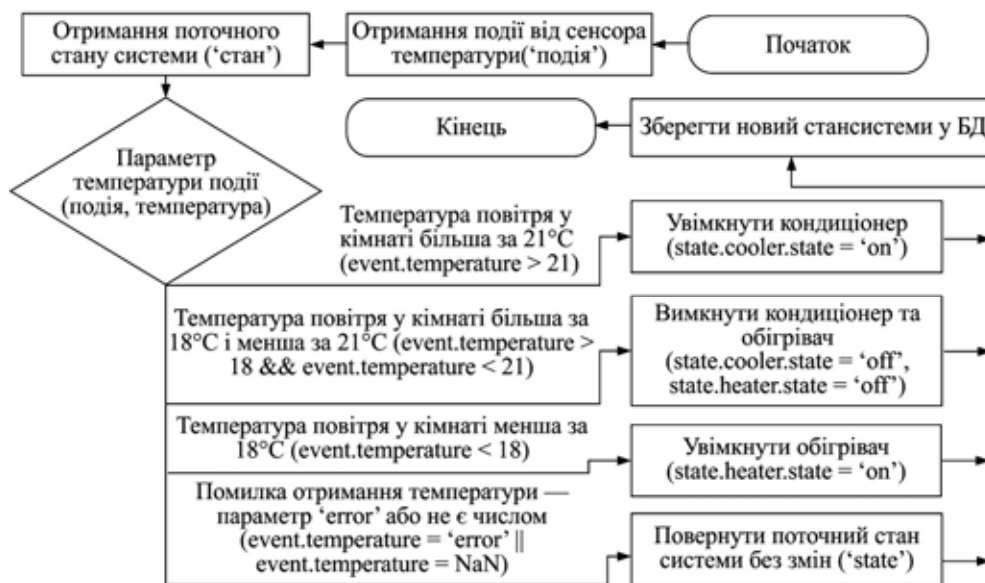


Рис. 3. Приклад алгоритму роботи редюсера

Переваги використання Redux у системах "розумного" будинку. Переваги використання обраного архітектурного рішення чітко зрозумілі на прикладі роботи СРБ з багатьма приміщеннями та сенсорами і побутовими приладами. Переваги використання Redux відобразимо за допомогою моніторингу швидкодії роботи однієї з розроблених СРБ, яку було проєктовано для офісу, що складається з двох кабінетів, приймальні, коридору, санвузла та великого робочого приміщення, поділеного на зони. Створена система отримує дані з датчиків руху і освітлення та керує освітлювальними приладами у всіх приміщеннях офісу за допомогою контролера, що побудований з використанням датчиків руху на базі одного мікрокомп'ютера Raspberry PI (Bakir, 2018; Jayakumar & Muthulakshmi, 2018). Датчики та освітлювальні прилади мають відповідний розподіл по приміщеннях (таблиця). За початкової реалізації системи було використано підхід прямої комунікації між датчиками, розташованими у приміщеннях офісу.

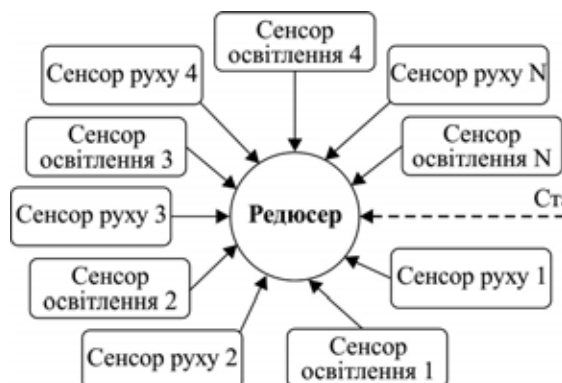
Таблиця. Розподіл датчиків та приладів системи "розумного" будинку по приміщеннях

Назва приміщення	Датчики руху	Датчики освітлення	Освітлювальні прилади
Кабінет 1	3	2	5
Кабінет 2	2	2	4
Приймальня	2	1	2
Коридор	3	2	3
Робоче приміщення. Зона 1	3	2	6
Робоче приміщення. Зона 2	2	1	5
Санвузол	1	1	3

Під час подальшого розширення офісних площ та додавання нових приміщень у керування системою, під час практичного використання цей підхід виявився складно-масштабованим та з очевидною тенденцією на зниження швидкодії роботи системи завдяки збільшенню кількості нових датчиків та побутових приладів, доданих у систему. Можливості масштабування та подальшого розвитку системи за використаного підходу взаємодії компонентів (датчики, прилади) системи обмежуються заплутаністю взаємозв'язків між компонентами

ми для зміни станів системи та повідомлення усіх решти компонентів про оновлення стану.

Під час використання архітектурного шаблону Redux давачі та прилади не взаємодіють безпосередньо між собою. Усі давачі надсилають події зміни своїх значень редюсеру, який змінює стан системи у СД відповідно до розробленої внутрішньої логіки реагування на



конкретну подію. Своєю чергою прилади підписані на "прослуховування" змін у сховищі даних та змінюють власні налаштування роботи під час виникнення зміни параметрів СС. Взаємозв'язки між усіма давачами та приладами в розробленій системі з використанням архітектурного шаблону Redux зображено на рис. 4.



Рис. 4. Діаграма зв'язків під час використання архітектурного шаблону Redux

Аналіз швидкодії роботи розглянутої системи "розумного" будинку відбувся на підставі випадково вибраної множини змін станів системи зі збереженими часовими мітками на кожному етапі отримання та надсилання команд між компонентами системи. Перша вибірка містить множину з 30 випадково відібраних записів змін станів завдяки використанню підходу безпосередньої взаємодії між компонентами системи. Друга вибірка містить множину з 30 випадково відібраних записів змін станів унаслідок запровадження архітектурного шаблону Redux.

Аналіз отриманих результатів роботи та порівняння показників відображає переваги використання архітектурного шаблону Redux у сфері СРБ. Такими перевагами є зменшення кількості запитів між різними компонентами системи, що пришвидшує швидкодію роботи системи та підвищує загальну надійність системи. Також завдяки зменшенню кількості взаємозв'язків між компонентами системи досягається пришвидшення зміни стану системи. Запровадження архітектурного шаблону Redux дало змогу зменшити кількість запитів між компонентами системи від 120 запитів до 60, що становить 50 % зменшення від початкового значення отриманого до модернізації системи. Швидкість оновлення загального стану системи зросла на 0,396 с, що становить 46 % пришвидшення роботи системи, порівняно з попередніми вимірами швидкості роботи системи до її модернізації.

Порівняння кількостей запитів між компонентами системи, що відбуваються у проектованій системі за прямої взаємодії компонентів системи між собою та під час використання архітектурного шаблону Redux, зображено на рис. 5.

Отримані результати дають змогу стверджувати, що запровадження архітектурного шаблону Redux зменшує кількість взаємозв'язків між компонентами системи на 50 %, що дає змогу збільшити швидкість оновлення стану системи та роботу системи на 46 %.

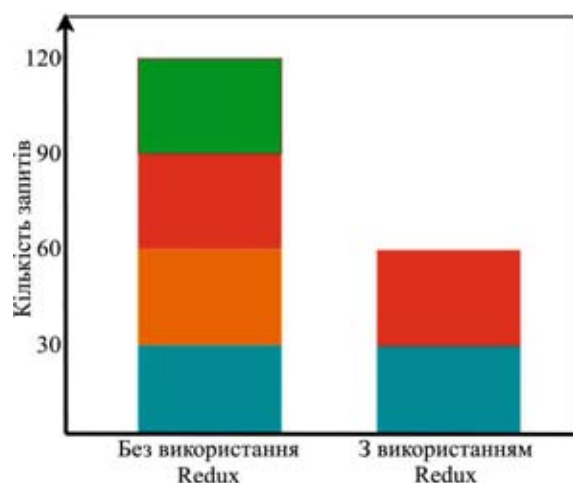


Рис. 5. Результати порівняння кількостей запитів між компонентами проектованої системи

Висновки. Отже, розроблена архітектура системи "розумного" будинку на підставі шаблону Redux надає централізований механізм управління станами даних системи під час роботи з великими масивами подій, що виникають у системах цього типу.

Розроблено метод, який базується на застосуванні архітектурного шаблону Redux з використанням централізованого сховища даних і дає змогу істотно знизити кількість взаємозв'язків між компонентами СРБ, що підвищує її надійність, а також показники швидкодії роботи системи загалом. Розроблено алгоритм реалізації методу та досліджено переваги використання розробленого алгоритму методу проектування систем "розумних" будинків з використанням архітектурного шаблону Redux на підставі порівняння показників швидкодії системи до та після впровадження запропонованого рішення.

Перелік використаних джерел

- Bakir, A. (2018). Setting Up a Raspberry Pi and Using It As a Home-Kit Bridge. In: *Program the Internet of Things with Swift for iOS*. Apress, Berkeley, CA, 235–266.
- Boreiko, O., Teslyuk, V., Zelinskyy, A., & Berezhsky, O. (2017). Development of models and means of the server part of the system for passenger traffic registration of public transport in the "smart" city.

- (Vol. 1). *Eastern-European Journal of Enterprise Technologies*, 2(85), 40–47.
- Cai, H., Xu, B., Jiang, L., & Vasilakos, A. V. (2017). IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenge. *IEEE Internet of Things Journal*, 4(1), 75–87.
- Christudas, B. (2019). Microservices in Depth. In *Practical Microservices Architectural Patterns*. Apress, Berkeley, CA, 35–53.
- Event-Based Systems. (2019). Engineering of Event-Based Systems. In: *Distributed Event-Based Systems*. Springer, Berlin, Heidelberg, 129–148.
- Familiar, B. (2015). From Monolithic to Microservice. In: *Microservices, IoT, and Azure*. Apress, Berkeley, CA, 1–7
- Freeman, A. (2019). Using a Redux Data Store. In: *Pro React 16*. Apress, Berkeley, CA, 531–559
- Gackenheim, C. (2015). Introducing Flux: An Application Architecture for React. In: *Introduction to React*. Apress, Berkeley, CA, 87–106.
- Islam Naim, N. (2017). ReactJS: An Open Source JavaScript Library for Front-end Development. *Metropolia University of Applied Sciences*. Retrieved from: https://www.theseus.fi/bitstream/handle/10024/130495/FInal_Year_Thesis.pdf?sequence=1&isAllowed=y.
- Jayakumar, A. J. K., & Muthulakshmi, S. (2018). Raspberry Pi-Based Surveillance System with IoT. In D. Thalmann, N. Subhashini, K. Mohanaprasad, M. Murugan (Eds). *Intelligent Embedded Systems. Lecture Notes in Electrical Engineering*, 492, 173–185. Singapore: Springer.
- Kalske, M., Mäkitalo, N., & Mikkonen, T. (2018). Challenges When Moving from Monolith to Microservice Architecture. In I. Garrigós, M. Wimmer (Eds), *Current Trends in Web Engineering. ICWE 2017. Lecture Notes in Computer Science*, 10544, 32–47. Cham: Springer.
- Kazarian, A., Teslyuk, V., Tsmots, I., & Mashevskaya, M. (2017). Units and structure of automated "smart" house system using machine learning algorithms. *Proceeding of the 14th International Conference " The Experience of Designing and Application of Cad Systems in Microelectronics*, CADSM2017, Polyana, Lviv, February 21–25, 2017. (pp. 364–366).
- Molnár, E., Molnár, R., Kryvinska, N., & Greguš, M. (2014). Web Intelligence in practice, The Society of Service Science. *Journal of Service Science Research*, 6(1), 149–172.
- Nene, A. V., Joseph, C. T., & Chandrasekaran, K. (2019). Construing Microservice Architectures: State-of-the-Art Algorithms and Research Issues. In L. Uden, I.H. Ting, J. Corchado (Eds). *Knowledge Management in Organizations. KMO 2019. Communications in Computer and Information Science*, 1027, 364–376. Cham: Springer.
- Paul, A., & Nalwaya, A. (2016). Flux: Solving Problems Differently. In: *React Native for iOS Development*. Apress, Berkeley, CA, 75–93
- Piispanen, M. (2017). *Modern architecture for large web applications*. Retrieved from: <https://jyx.jyu.fi/bitstream/handle/123456789/54129/1/URN%3ANBN%3Afi%3Aju-201705272524.pdf>.
- Poniszewska-Maranda, A., Kaczmarek, D., Kryvinska, N., et al. (2018). Studying usability of AI in the IoT systems/paradigm through embedding NN techniques into mobile smart service system. *Computing*, 1–25.
- Saransig, A., & Tapia, F. (2019). Performance Analysis of Monolithic and Micro Service Architectures – Containers Technology. In: Mejia J., Muñoz M., Rocha Á., Peña A., Pérez-Cisneros M. (Eds) *Trends and Applications in Software Engineering. CIMPS 2018. Advances in Intelligent Systems and Computing*, vol. 865. Springer, Cham, 270–279.
- Sultan, M., & Ahmed, K. N. (2017). SLASH: Self-learning and adaptive smart home framework by integrating IoT with big data analytics. *Computing Conference*, London. (pp. 530–538).
- Teslyuk, T., Tsmots, I., Teslyuk, V., Medykovskyy, M., & Opotyak, Y. (2018). Architecture and Models for System-Level Computer-Aided Design of the Management System of Energy Efficiency of Technological Processes at the Enterprise. In N. Shakhovska, V. Stepashko (Eds), *Advances in Intelligent Systems and Computing II. CSIT 2017. Advances in Intelligent Systems and Computing* 689, 538–557. Cham: Springer.
- Teslyuk, V., Denysyuk, P., Al Shawabkeh, H. A. Y., & Kernyskyk, A. (2010). Developing the information model of the reachability graph. *Proc. of the 15th International Seminar/Workshop on Direct and Inverse Problems of Electromagnetic and Acoustic Wave Theory, DIPED'2010*, Tbilisi, September 27–30. (pp. 210–214).

V. M. Teslyuk, I. H. Tsmots, A. G. Kazarian, T. V. Teslyuk
Lviv Polytechnic National University, Lviv, Ukraine

THE METHOD OF SMART HOME SYSTEMS DESIGN USING THE REDUX ARCHITECTURAL PATTERN

Article presents the method for designing of smart home systems using the Redux architectural template. The method is based on the adaptation of the Redux architectural template usually used in designing of visual interfaces for use in the Internet of Things sphere. The system of smart home for control of lighting devices with the help of motion and lighting sensors in the rooms of office building was constructed, based on the developed method. The developed design method allows increasing the system reliability and performance. Improved reliability is achieved by reducing the number of direct relationships between system components. The developed design method also helps reduce the amount of information that is duplicated in different components of the system by using one common data store to save the state, which increases the speed of updating the state of the system and the speed of lighting appliances settings changing. The benefits of using the developed design method are experimentally demonstrated by emulating the work of the smart home system, with saving and analysis of lighting settings change time, before and after the usage of Redux pattern. The design method for smart home systems using the Redux architectural pattern allows scaling system by adding new sensors and appliances to the developed system without losing the speed of data processing and transmission of control commands to the devices. The example presented in this paper show the advantage of developed method usage for designing of smart home systems, that will provide the functionality of appliances automated control in large residential, administrative and office buildings with a large number of simultaneously occurring events that are detected by system for further processing and require appropriate changes to the state of the smart home system settings.

Keywords: design; architectural template; Redux; smart home.