

УДК 004.514

*Кодола Г.М., Венедіктов Д.В.***ПРО МЕТОДИ ДИНАМІЧНОЇ ПОБУДОВИ ГРАФІЧНОГО ІНТЕРФЕЙСУ
КОРИСТУВАЧА НА ОСНОВІ СТРУКТУРИ ПРИКЛАДНОЇ ОНТОЛОГІЇ****ДВНЗ «Український державний хіміко-технологічний університет», м. Дніпро, Україна**

Сучасні засоби для розробки користувацького інтерфейсу, а саме – будівники інтерфейсу і системи управління користувацьким інтерфейсом пропонують більш простий, зручний і зрозумілий підхід до розробки інтерфейсу користувача. Для реалізації основних вимог сучасного світу до розробки програмних засобів, а зокрема і графічного користувацького інтерфейсу запропонований підхід до розробки інтерфейсу, який полягає в динамічній побудові інтерфейсу користувача за допомогою загальнодоступних (opensource) технологій, в подальшому дозволить з легкістю модифікувати, а також підтримувати проект. Для вирішення проблеми динамічної побудови графічного інтерфейсу пропонується підхід до розробки інтерфейсу, заснований на онтологіях. Основна ідея підходу – формувати декларативну модель користувацького інтерфейсу на основі універсальних моделей онтологій і потім за високорівневим декларативним описом автоматично генерувати виконавчий код інтерфейсу. Для спрощення формування компоненти моделі інтерфейсу – запропонована модель онтології «графічний користувацький інтерфейс», яка описує графічні інтерфейсні елементи, їх властивості та зв'язок один з одним для формування діалогу з користувачем, використовуючи систему понять користувача, яку можна розділити на систему понять діалогу і систему понять завдань користувача. Основними елементами інтерфейсу, які призначені для угруповання в пов'язані групи і класи, є елементи типу «вікно-контейнер», елементи для операцій введення/виведення даних і виклику команд (елементи управління), елементи для опису. Кожен параметр елемента інтерфейсу складається з імені і типу його можливого значення. З кожним елементом інтерфейсу зв'язується безліччю його подій, які визначають реакції елемента інтерфейсу при його взаємодії з користувачем. Для опису безлічі можливих дій над елементом інтерфейсу використовуються функції. Запропонований підхід до розробки інтерфейсу заснований на онтологіях дозволить просто і зручно будувати інтерфейс користувача на веб-сторінці, що значно прискорить час на розробку даного етапу в реалізації проекту.

Ключові слова: графічний інтерфейс веб-додатку, прикладна онтологія, модель онтології, інтерфейсний елемент, параметри елемента інтерфейсу.

DOI: 10.32434/2521-6406-2018-4-2-27-33

Постановка проблеми

Розробка інформаційних систем – складне завдання, що вимагає від розробників знань в різних областях: проектування баз даних, програмування логіки роботи системи, реалізація інтерфейсу та ін. Складність розробки ще більш зростає, якщо розглядати веб-орієнтовані системи. Це пов'язано із застосуванням різних мов і засобів розробки на різних рівнях. Зазвичай для спрощення розробки використовуються

стандартні методології, архітектури та набори готових компонентів, реалізованих у вигляді «програмних каркасів» (framework). Тим не менш, при створенні подібних систем часто потрібно повторне написання коду, що реалізує схожий функціонал. Повторне кодування схожих функцій часто потрібно навіть при створенні однієї системи, наприклад, якщо мається на увазі управління однотипними об'єктами. В якості теоретичної бази для побудови таких мо-

делей об'єктів передбачається застосовувати онтології.

Аналіз останніх досліджень і публікацій

Аналіз досліджень в даній області показав актуальність питань використання онтологій при розробці графічного інтерфейсу користувача в різноманітних предметних областях.

Зокрема, в роботі [1] Локтева Д.А. і Локтева А.А. надано проектування та реалізація на основі онтологічного підходу людино-машинного інтерфейсу, який пов'язує всі програмні модулі комплексної системи відеомоніторингу і розпізнавання об'єктів в єдиний комплекс.

В роботі [2] Грегера С.Е. та Поршнева С.В. розглядаються підходи до створення єдиної онтології архітектури інформаційної системи (ІС), засновані на методах і технологіях управління знаннями, а також спосіб її використання для узгодження функціональних і структурних описів цільової і забезпечуючих систем.

На даний час активно обговорюються проблеми використання методів семантичного підходу для узгодження описів взаємодіючих систем та способи технічної реалізації таких семантичних моделей [3–5]. Проблема онтології і використання їх в комп'ютерних системах розглядалася В. Лапшиним [6]. Детально процес розробки онтології в загальному вигляді розглянуто в роботі дослідників Н. Ной і Д. Мак Гіннеса [7]. Палагіним О.В. та Петренко М.Г. був складений тлумачний словник з інженерії знань [8], який «семантизував» терміни, які активно використовуються в ряді близьких наукових дисциплін з комп'ютерної та програмної інженерії, у вигляді онтологічного графа з кластерним розбиттям змісту його вершин.

Однак ціла низка питань, пов'язаних з практичною реалізацією зазначених підходів, як і раніше, залишається актуальною.

Формулювання цілей статті

Розглянути теоретичні основи побудови моделей об'єктів динамічної розробки графічного інтерфейсу користувача на веб-сторінці на основі структури прикладної онтології. Описати модель побудови графічного інтерфейсу користувача для веб-додатку.

Вклад основного матеріалу дослідження

Онтологією називається спосіб формалізації предметних областей у вигляді понятійної (концептуальної) схеми, тобто системи взаємопов'язаних понять, а якщо бути більш конкретним, то онтологія – це артефакт, структура, що описує значення елементів деякої системи. На зразок класів в об'єктно-орієнтованому програму-

ванні (ООП), поняття можуть успадковуватися один від одного, при цьому обов'язково підтримується множинне спадкування, відсутнє в багатьох об'єктно-орієнтованих мовах програмування. Кожне поняття визначається набором властивостей, які повинні мати екземпляри даного поняття. Одна з відмінностей від ООП тут полягає в тому, що властивості можуть визначатися окремо від понять і поняття потім можуть складатися з наявних властивостей. Таким чином, в процесі функціонування програми шляхом комбінування відомих властивостей можуть створюватися нові поняття, які не визначені в вихідній онтології. Окремі примірники також можуть мати довільний набір властивостей і потім за допомогою процедур виведення бути віднесені до того чи іншого поняття. Підтримка процедур виведення є основною особливістю онтологій, саме на цю особливість необхідно орієнтуватися при розробці графічного інтерфейсу [9].

Сучасні засоби для розробки користувальницького інтерфейсу, а саме – будівники інтерфейсу і системи управління користувальницьким інтерфейсом пропонують більш простий, зручний і зрозумілий підхід до розробки інтерфейсу користувача.

Для реалізації основних вимог сучасного світу до розробки програмних засобів, а зокрема і графічного користувальницького інтерфейсу (ГК – графічний інтерфейс користувача) запропонований підхід до розробки інтерфейсу, який полягає в динамічній побудові інтерфейсу користувача за допомогою загальнодоступних (opensource) технологій, що в подальшому дозволить з легкістю модифікувати, а також підтримувати проект. В основі методу розробки лежить простота в застосуванні, що дозволить розробнику швидко і просто зрозуміти принцип роботи даного програмного забезпечення. Основними положеннями концепції розробки динамічної побудови інтерфейсу є наступні концепції [3]:

- об'єднання однорідних за структурою елементів сторінки в контейнери;
- можливість генерації елементів з різними параметрами, наприклад, ширини, висоти та інше;
- можливість додавання наповнення структурного елемента, наприклад, параграфи або заголовки;
- властивість позиціонування наповнення в залежності від батьківського елемента (елемент всередині якого знаходиться наповнювач), наприклад, позиціонування тексту по центру;

– властивість позиціонування безпосередньо елемента за допомогою курсору в будь-якій ділянці сторінки;

– можливість збереження цієї структури для подальшого використання в особистих цілях.

Для динамічної побудови графічного інтерфейсу можуть бути використані такі технології як: Angular 6, React, VueJs, Knockoutjs, Backbonejs, jQuery, jQueryUI.

На відміну від статичної побудови у розробника не буде необхідності прописувати тисячі рядків коду для створення власної теми або індивідуального дизайну, а щоб не витратити багато часу на його розташування в динамічній побудові графічного інтерфейсу буде включена функція перетягування і можливість змінювати його розмірність безпосередньо на екрані, для більшого візуального сприйняття.

Для вирішення проблеми динамічної побудови графічного інтерфейсу пропонується підхід до розробки інтерфейсу, заснований на онтологіях.

Основна ідея підходу – формувати декларативну модель користувальницького інтерфейсу на основі універсальних моделей онтологій і потім за високорівневим декларативним описом автоматично генерувати виконавчий код інтерфейсу. Для спрощення формування компонента моделі інтерфейсу – моделі виразних засобів – розроблена модель онтології «графічний користувальницький інтерфейс», яка описує графічні інтерфейсні елементи, їх властивості та зв'язок один з одним для формування діалогу з користувачем, заснованому на екранних формах [10].

Підхід для побудови ГІК, який заснований на використанні онтології для надання використовуваних понять, дозволяє більш детально описати проєктований користувальницький інтерфейс для подальшого надання в машинному коді. Для застосування такого підходу необхідно сформулювати систему понять користувача, яку можна розділити на систему понять діалогу і систему понять задач користувача.

Система понять діалогу – це система понять, через яку виражаються вхідні і вихідні дані, здійснюється інтелектуальна підтримка користувача в процесі його взаємодії з програмним засобом.

Система понять задач користувача описує задачі, які він може вирішити, використовуючи програмний засіб, при цьому окрема задача може складатися з набору підзадач. Згідно з онтологічним підходом кожен елемент WIMP-інтерфей-

су визначається своїм типом, безліччю параметрів, подій і функцій.

Онтологічний підхід включає в себе побудову кількох моделей онтологій. Існують наступні моделі онтологій [11]:

– модель онтології системи понять діалогу (ОСПД) описує структуру термінів системи понять діалогу у вигляді зв'язків між ними;

– модель онтології задач користувача (ОЗК) описує задачі, які вона може вирішити за допомогою програмного засобу;

– модель онтології WIMP-інтерфейсів складається з двох рівнів: метаонтології, призначеної для опису структури інтерфейсних елементів і безпосередньо онтології, що містить опис безлічі інтерфейсних елементів відповідно до структури, представленої в метаонтології.

– модель метаонтології WIMP-інтерфейсів (ОПР) – це безліч елементів інтерфейсу (Controls), що пов'язані з подіями (Events), які задають безліч можливих реакцій.

Також необхідно ввести поняття «звернення до елемента знань». Це поняття охоплює ті дії з елементами онтології, які будуть вважатися ключовими для запам'ятовування системою часу.

Будемо вважати, що має місце звернення до елемента знань, якщо користувач здійснив такі дії:

– створення елемента;

– додавання елемента в уявлення;

– перегляд (деталізація) елемента;

– редагування елемента;

– побудова відносин між даним елементом і іншим елементом (елементами).

Нещодавно додані, переглянуті або відредаговані елементи можуть виділятися графічно, наприклад, за рахунок упорядкування, різної інтенсивності фарбування або розміру.

Структурне відображення вищенаведених онтологічних моделей [12]:

ОСПД= \langle Ім'я Системи Понять, Групи Термінів \rangle ,

де Ім'я Системи Понять – ім'я системи понять діалогу;

Групи Термінів – безліч груп термінів:

Групи Термінів={Група Термінів}.

Модель онтології задач користувача (ОЗК) описує задачі, які вона може вирішити за допомогою програмного засобу:

ОЗК= \langle Ім'я Загальної Задачі, Задачі \rangle ,

де Ім'я Загальної Задачі – ім'я спільної задачі;

Задачі – задачі користувача:

Задачі= \langle Тип Множини, {Задача \rangle ,

де Тип Множини – тип безлічі,
Задача – задача користувача.

Основними елементами інтерфейсу, призначеними для угруповання в пов'язані групи і класи, є елементи типу «вікно-контейнер», елементи для операцій введення/виведення даних і виклику команд (елементи управління), елементи для опису. Кожен параметр елемента інтерфейсу складається з імені і типу його можливого значення. З кожним елементом інтерфейсу зв'язується безліччю його подій, які визначають реакції елемента інтерфейсу при його взаємодії з користувачем. Для опису безлічі можливих дій над елементом інтерфейсу використовуються функції [13].

Серед різних видів візуальних засобів графічного інтерфейсу, призначеного для користувача, виділяються дві основні групи – вікна та віконні елементи управління, і три додаткові – панелі управління, віконні меню і допоміжні засоби. Фрагмент ієрархії елементів ГІК (класів) представлений на рисунку.

Елемент ГІК – клас, що описує загальні для всіх елементів ГІК властивості: X (екранна координата лівого верхнього кута елемента по осі X в пікселях, тип: ціле число); Y (екранна координата лівого верхнього кута елемента по осі Y в пікселях, тип: ціле число); ширина (ши-

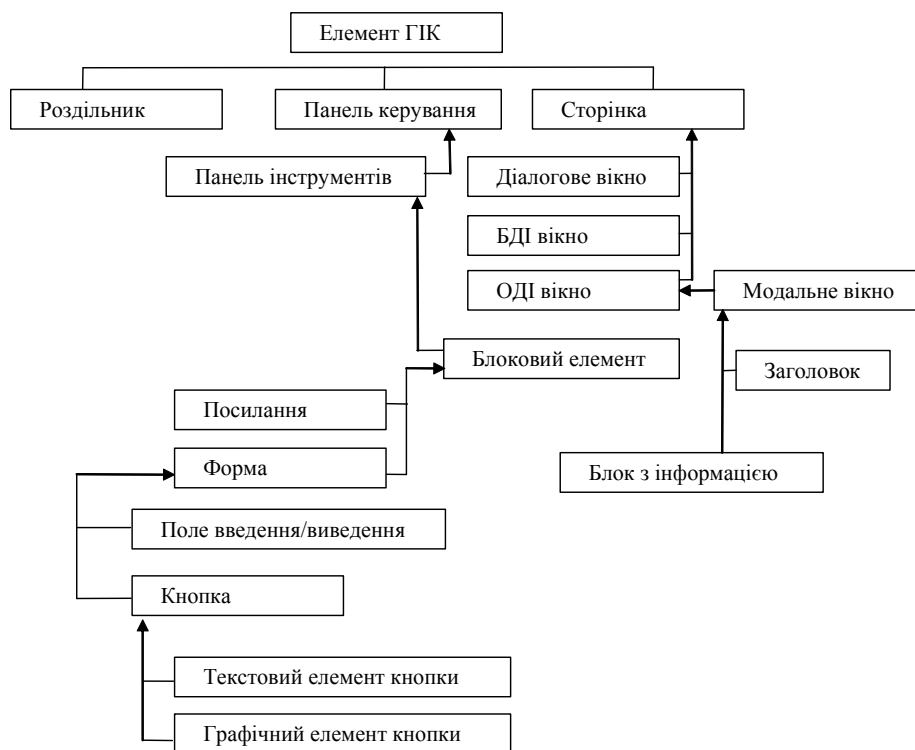
рина елемента в пікселях, тип: ціле число); висота (висота елемента в пікселях, тип: ціле число); відображення (ознака, чи відображається елемент на екрані, тип: булевський); доступність (ознака, чи доступний елемент для взаємодії з користувачем, тип: булевський); меню (кожному елементу відповідає контекстне меню у вигляді блоку даних, тип: блок даних) та ін.

ГІК елемент – основний елемент структури, який може бути використаний в онтологічному підході до розробки графічного інтерфейсу.

При розробці використовують різні моделі побудови графічного інтерфейсу, найбільш відомі з них це:

- модель однодокументного інтерфейсу (ОДІ);
- модель багатодокументного інтерфейсу (БДІ);
- модель діалогового інтерфейсу.

Серед різних видів візуальних засобів інтерфейсу, призначеного для користувача, виділяються дві основні групи – вікна та віконні елементи управління, тому що програмне забезпечення буде використовуватися для розробки графічного інтерфейсу на веб-сайтах, вікна в даному контексті будуть описуватися як блоки, а віконні елементи як блокові.



Фрагмент ієрархії елементів ГІК

Блоки є головними об'єктами в інтерфейсі. Вони структурують інтерфейс, призначений для користувача. Введення-виведення даних для кожної окремої підзадачі здійснюється у відповідному блоці. Всі інші об'єкти інтерфейсу можуть існувати тільки в якості складової частини будь-якого блоку.

Блокові елементи управління розташовуються усередині блоку. За їх допомогою користувач і здійснює безпосередню взаємодію з веб-сторінкою. Приклади блокових елементів: кнопки, поля введення, статичний текст, списки, форми та ін.

Так само існують панелі управління, які представляють собою області, що містять інтерфейсні елементи, як правило, кнопки, які прикріплюються до однієї з внутрішніх сторін вікна, або поміщаються поверх нього. Панелі управління слугують для швидкого виклику команд програми. У більшості інтерфейсів за функціями вони дублюють інші, більш потужні, але в той же час більш «повільні» засоби взаємодії з користувачем.

На панелях управління розміщуються засоби доступу до найбільш часто вживаних елементів структури або команд.

Роздільники – це елементи інтерфейсу, призначеного для користувача, які розташовуються усередині сторінки і слугують для розбиття його на самостійні частини, з кожною з яких користувач може взаємодіяти як з самостійним елементом структури. Роздільник розбиває вікно на декілька частин, організованих на екрані в будь-якій зручній користувачеві формі. Кожен такий елемент може містити власні віконні елементи управління. Роздільники в основному використовуються тільки в багатосторінковому інтерфейсі.

Окремо можна виділити модальні вікна, в яких може бути присутнім підказка, форма, необхідна для розуміння інформація, в деяких випадках такі вікна називають вікно з довідковою інформацією. Їх відображення відбувається при наведенні покажчика миші на інтерфейсний елемент і зникає при переміщенні покажчика на інший інтерфейсний елемент, або при натисканні на який-небудь елемент сторінки з реалізованим функціоналом, а закриття при натисканні на відповідну кнопку.

Висновки

За допомогою перерахованих вище технологій, понять і методологій, а також їх зв'язку з елементами об'єктної моделі документа, можлива проста і зручна реалізація динамічної по-

будови інтерфейсу користувача на веб-сторінці, що значно прискорить час на розробку даного етапу в реалізації проекту.

СПИСОК ЛІТЕРАТУРИ

1. Loktev D.A., Loktev A.A. Development of a User Interface for an Integrated System of Video Monitoring Based on Ontologies // Contemporary Engineering Sciences. – 2015. – Vol.8 – № 17. – PP.789-797.
2. Грегер С.Э., Поршнев С.В. Построение онтологии архитектуры информационной системы // Фундаментальные исследования. – 2013. – № 10-11. – С.2405-2409.
3. Guarino N. Formal Ontology in Information Systems // Formal Ontology in Information Systems. Proceedings of FOIS'98 (Trento, Italy, June 6-8, 1998). – Amsterdam: IOS Press. – PP.3-15.
4. Diepenbrock A., Rademacher F., Sachweh S. An Ontology-based Approach for Domain-driven Design of Microservice Architectures // In: Eibl, M. & Gaedke, M. (Hrsg.), INFORMATIK 2017. Gesellschaft für Informatik, Bonn. – 2017. – PP.1777-1791.
5. An ontological foundation for conceptual modeling datatypes based on semantic reference spaces / Albuquerque, Antognoni; Guizzardi, Giancarlo // In: 7th International Conference on Research Challenges in Information Science (RCIS 2013), IEEE. – 2013. – PP.1-12.
6. Лапшин В.А. Онтологии в компьютерных системах [Электронный ресурс]. – Режим доступа: <http://www.rsdn.ru/article/philosophy/what-is-onto.xml>.
7. Noy N., McGuinness D. Ontology Development 101: A Guide to Creating Your First Ontology. – Stanford Knowledge Systems Laboratory Technical Report KSL – 01-05 and Stanford Medical Informatics Technical Report SMI – 2001-0880, March 2001 [Электронный ресурс]. – Режим доступа: [http://protege.stanford.edu/publications/ontology development/ontology101.pdf](http://protege.stanford.edu/publications/ontology%20development/ontology101.pdf).
8. Тлумачний онтографічний словник з інженерії знань / Скл. Палагін О.В., Петренко М.Г. – Київ: ТОВ «НВП Інтерсервіс», 2017. – 478 с.
9. Mark Green, Robert Jacob. SIGGRAPH: 90 Workshop report: software architectures and metaphors for non-WIMP user interfaces // ACM SIGGRAPH Computer Graphics. – July 1991. – № 25(3). – PP.229-235.
10. Gruber T. Toward. Principles for the Design of Ontologies Used for Knowledge Sharing // International Journal of Human-Computer Studies. – November 1995. – Vol.43. Issues 5-6. – PP.907-928.
11. Nirenburg S., Raskin V. Ontological Semantics. – Cambridge MA. – 2004. – 332 p.
12. Gribova V.V., Tarasov A.V. The ontology model of the «Graphical User Interface» domain // [Intelligent system], Intellektual'nye Sistemy. – 2005. – Issue 1(9).

13. *Ontology visualization methods – a survey* / Katifori A., Halatsis C., Lepouras G. et. – ACM Comput. Surv. – 2007. – Vol.39. – № 4. – 38 p.

Надійшла до редакції 22.10.2018

О МЕТОДАХ ДИНАМИЧЕСКОГО ПОСТРОЕНИЯ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ НА ОСНОВЕ СТРУКТУРЫ ПРИКЛАДНОЙ ОНТОЛОГИИ

Кодола Г.Н., Венедиктов Д.В.

Современные средства для разработки пользовательского интерфейса, а именно – построители интерфейса и системы управления пользовательским интерфейсом, предлагают более простой, удобный и понятный подход к разработке пользовательского интерфейса. Для реализации основных требований современного мира к разработке программных средств, а в частности и графического пользовательского интерфейса предложенный подход к разработке интерфейса, который заключается в динамическом построении интерфейса с помощью общедоступных (opensource) технологий, в дальнейшем позволит с легкостью модифицировать, а также поддерживать проект. Для решения проблемы динамического построения графического интерфейса предлагается подход к разработке интерфейса, основанный на онтологиях. Основная идея подхода – формировать декларативную модель пользовательского интерфейса на основе универсальных моделей онтологий и затем по высокоуровневому декларативному описанию автоматически генерировать исполнимый код интерфейса. Для упрощения формирования компоненты модели интерфейса – предложена модель онтологии «графический пользовательский интерфейс», которая описывает графические интерфейсные элементы, их свойства и связь друг с другом для формирования диалога с пользователем, используя систему понятий пользователя, которую можно разделить на систему понятий диалога и систему понятий задач пользователя. Основными элементами интерфейса, предназначенными для группировки в связанные группы и классы, есть элементы типа «окно-контейнер», элементы для операций ввода/вывода данных и вызова команд (элементы управления), элементы для описания. Каждый параметр элемента интерфейса состоит из имени и типа его возможного значения. С каждым элементом интерфейса связывается множеством его событий, которые определяют реакции элемента интерфейса при его взаимодействии с пользователем. Для описания множества возможных действий над элементом интерфейса используются функции. Предложенный подход к разработке интерфейса, основанный на онтологиях, позволит просто и удобно строить интерфейс пользователя на веб-странице, что значительно ускорит время на выполнение данного этапа в реализации проекта.

Ключевые слова: графический интерфейс веб-приложения, прикладная онтология, модель онтологии, интерфейсный элемент, параметры элемента интерфейса.

ABOUT METHODS OF DYNAMICALLY CONSTRUCTING GRAPHICAL USER INTERFACE BASED ON THE STRUCTURE OF APPLIED ONTOLOGY

Kodola G.N., Veniediktov D.V.

Ukrainian State University of Chemical Technology, Dnipro, Ukraine

Modern tools for developing user interfaces, namely, interface builders and user interface management systems, offer a simpler, more convenient and understandable approach to developing user interfaces. To implement the basic requirements of the modern world for software development and, in particular, for development of graphical user interface, the proposed approach to interface development, which consists in dynamically building an interface using publicly available (opensource) technologies, will make it possible to easily modify and maintain the project in the future. To solve the problem of dynamic graphical interface building, an approach to interface development based on ontologies is proposed. The main idea of the approach is to form a declarative user interface model based on universal ontology models and then automatically generate the executable interface code using a high-level declarative description. To simplify the formation of the interface model components, the ontology model “graphical user interface” is proposed, which describes the graphical interface elements, their properties and communication with each other to form a dialogue with the user, using the user concept system, which can be divided into the dialogue concept system and the concept system user tasks. The main interface elements designed to be grouped into related groups and classes are window-container elements, elements for data input/output operations and commands (controls), elements for description. Each parameter of an interface element consists of a name and the type of its possible value. Each interface element is associated with a set of its events that determine the reactions of an interface element when it interacts with the user. To describe the set of possible actions on an interface element, functions are used. The proposed approach to the development of an interface based on ontologies will make it easy and convenient to build a user interface on a web page, which will significantly speed up the time to complete this stage in the project implementation.

Keywords: web-application graphical interface, application ontology, ontology model, interface element, interface element parameters.

REFERENCES

1. Loktev D.A., Loktev A.A. Development of a User Interface for an Integrated System of Video Monitoring Based on Ontologies. Contemporary Engineering Sciences, 2015, vol.8, no. 17, pp.789-797.
2. Greger S.E., Porshnev S.V. Postroenie ontologii arhitektury informatsionnoy sistemy [Construction of the ontology of the architecture of the information system] Fundamentalnyye issledovaniya [Fundamental research]. 2013, no 10-11, pp.2405-2409. (in Russian).
3. Guarino N. Formal Ontology in Information Systems. Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, June 6-8, 1998, Amsterdam: IOS Press, pp.3-15.
4. Diepenbrock A., Rademacher F., Sachweh S. An Ontology-based Approach for Domain-driven Design of Microservice Architectures. In: Eibl, M. & Gaedke, M. (Hrsg.), INFOR-MATIK 2017, Gesellschaft für Informatik, Bonn, 2017, pp.1777-1791.
5. Albuquerque, Antognoni; Guizzardi, Giancarlo An ontological foundation for conceptual modeling datatypes based on semantic reference spaces. IEEE In: 7th International Conference on Research Challenges in Information Science (RCIS 2013), 2013, pp.1-12.
6. Lapshin V.A. Ontologii v kompyuternykh sistemakh [Ontologies in computer systems] [SoftMetaWare] URL: <http://www.rsdn.ru/article/philosophy/what-is-onto.xml>. (in Russian).
7. Noy N., McGuinness D. Ontology Development 101: A Guide to Creating Your First Ontology. – Stanford Knowledge Systems Laboratory Technical Report KSL – 01-05 and Stanford Medical Informatics Technical Report SMI – 2001-0880, March 2001 [SoftMetaWare], URL: http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.
8. Tlumachnyj ontografichnyj slovnyk z inzheneriyi znan. [An Ontographic Dictionary of Experimental Knowledge Engineering.] Skl. Palagin O.V., Petrenko M.G. Kiev: LLC “NVP Interservis”, 2017, 478 p. (in Ukrainian)
9. Mark Green, Robert Jacob. SIGGRAPH: 90 Workshop report: software architectures and metaphors for non-WIMP user interfaces. ACM SIGGRAPH Computer Graphics, July 1991, no. 25(3), pp. 229-235.
10. Gruber T. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies, November 1995, vol.43, Issues 5-6, pp.907-928.
11. Nirenburg S., Raskin V. Ontological Semantics. Cambridge, MA, 2004, 332 p.
12. Gribova V.V., Tarasov A.V. The ontology model of the «Graphical User Interface» domain. [Intelligent system], Intellektual'nye Sistemy, 2005, Issue 1(9).
13. Katifori A., Halatsis C., Lepouras G. et. Ontology visualization methods – a survey. ACM Comput. Surv, 2007, vol.39, no. 4, 38 p.