

УДК 004.04

## Побудова робочих процесів на основі онтологій

Л.С. Глоба, В.В. Гаєвий, А.М. Бакай

**Глоба  
Лариса Сергіївна**

*д.т.н., професор; завідувачка кафедрою інформаційно-телекомунікаційних систем*

*Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,  
проспект Перемоги 37, Київ-56, Україна, 03056  
e-mail: [lgloba@its.kpi.ua](mailto:lgloba@its.kpi.ua);*

<https://orcid.org/0000-0003-3231-3012>

**Гаєвий  
Вадим Володимирович**

*аспірант*

*Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,  
проспект Перемоги 37, Київ-56, Україна, 03056  
e-mail: [vadim.gayevoy@outlook.com](mailto:vadim.gayevoy@outlook.com)*

<https://orcid.org/0000-0001-7855-3957>

**Бакай  
Андрій Миколайович**

*студент*

*Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»,  
проспект Перемоги 37, Київ-56, Україна, 03056  
e-mail: [andruha.bakay1997@gmail.com](mailto:andruha.bakay1997@gmail.com)*

Сучасне суспільство є неможливим без цифрової економіки, яка все більше надає значний обсяг послуг, гарантуючи технологічну ефективність та інновації, постійно оновлюючи технології та сервіси, адаптуючи їх до потреб кінцевого користувача, реконфігуруючи складні програмні системи. Все це вимагає постійного реінженірингу бізнес-процесів в предметних областях, а відповідно підвищення рівня їх автоматизації. Разом з тим, ефективних рішень, які б дозволили виконувати автоматизований реінженіринг робочих процесів, на даний момент не існує. У статті представлений підхід до побудови обчислювальних робочих процесів, який заснований на чітко визначених онтологічних моделях предметної області, онтологічній моделі набору правил побудови зв'язків між функціональними сервісами, і онтологічній моделі самих робочих процесів, що динамічно генеруються. Проведено експериментальні дослідження даного підходу та розроблено прототип інструментарію автоматизованої побудови обчислювальних робочих процесів. Застосування запропонованого підходу дозволить підвищити рівень автоматизації побудови обчислювальних робочих процесів, генерувати робочі процеси "на льоту" базуючись на заздалегідь підготовленому пулі функціональних сервісів і обмежень, накладених онтологією предметної області.

**Ключові слова:** онтологія, обчислювальні робочі процеси, функціональні сервіси, автоматизація робочих процесів.

## Development of workflows based on ontologies

**Globa Larisa**

*Doctor of Technical Sciences, Professor; Head of the Department of Information and Telecommunication Systems*

*National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Ave., Kyiv-56, Ukraine, 03056*

*e-mail: [lgloba@its.kpi.ua](mailto:lgloba@its.kpi.ua);*

<https://orcid.org/0000-0003-3231-3012>

**Gaievyi Vadym**

*PhD student*

*National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Ave., Kyiv-56, Ukraine, 03056*

*e-mail: [vadim.gayevoy@outlook.com](mailto:vadim.gayevoy@outlook.com)*

<https://orcid.org/0000-0001-7855-3957>

**Bakai Andrii**

*student*

*National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", 37 Peremohy Ave., Kyiv-56, Ukraine, 03056*

*e-mail: [andruha.bakay1997@gmail.com](mailto:andruha.bakay1997@gmail.com)*

Modern society is impossible without the digital economy, which ensures an ever-increasing number of services and guarantee technological efficiency and innovation, constantly updating technologies and services, adapting them to user needs, and reconfiguring complex software systems. That requires constant reengineering of the workflows for the subject areas, and, accordingly, an increase in the level of the automation. At the same time, currently there are no efficient solutions for automated reengineering of the workflows. This article presents an approach to the computational creation of the workflows based on well-defined ontological models of the domain, an ontological model of a set of rules for building connections between functional services, and an ontological model of the workflows that are generated dynamically. The research of this approach has been carried out and the prototype of the tools for the automated creation of the computational workflows has been developed. Using the approach proposed will increase the level of automation of the construction of the computational workflows, and allow generating workflows "on the fly" basing on a previously prepared pool of functional services and restrictions imposed by the ontology domain.

**Keywords:** *ontology, computational workflows, functional services, workflow's automation.*

## Построение рабочих процессов на основе онтологий

**Глоба**

**Лариса Сергеевна**

*д.т.н., профессор; заведующий кафедрой информационно-телекоммуникационных систем*

*Национальный технический университет Украины "Киевский политехнический институт имени Игоря Сикорского", проспект Победы 37, Киев-56, Украина, 0306*

*e-mail: [lgloba@its.kpi.ua](mailto:lgloba@its.kpi.ua);*

*<https://orcid.org/0000-0003-3231-3012>*

**Гаевой**

**Вадим Владимирович**

*аспирант,*

*Национальный технический университет Украины "Киевский политехнический институт имени Игоря Сикорского", проспект Победы 37, Киев-56, Украина, 03056*

*e-mail: [yadim.gayevoy@outlook.com](mailto:yadim.gayevoy@outlook.com)*

*<https://orcid.org/0000-0001-7855-3957>*

**Бакай**

**Андрей Николаевич**

*студент,*

*Национальный технический университет Украины "Киевский политехнический институт имени Игоря Сикорского", проспект Победы 37, Киев-56, Украина, 03056*

*e-mail: [andruha.bakay1997@gmail.com](mailto:andruha.bakay1997@gmail.com)*

Современное общество невозможно без цифровой экономики, которая все больше оказывает значительный объем услуг, гарантируя технологическую эффективность и инновации, постоянно обновляя технологии и сервисы, адаптируя их к потребностям конечного пользователя, реконфигурируя сложные программные системы. Все это требует постоянного реинжиниринга бизнес-процессов в предметных областях, а соответственно повышения уровня их автоматизации. Вместе с тем, эффективных решений, позволяющих выполнять автоматизированный реинжиниринг рабочих процессов, на данный момент не существует. В статье представлен подход к построению вычислительных рабочих процессов, который основан на четко определенных онтологических моделях предметной области, онтологической модели набора правил построения связей между функциональными сервисами, и онтологической модели самых рабочих процессов, динамически генерируемые. Проведены экспериментальные исследования данного подхода и разработан прототип инструментария автоматизированного построения вычислительных рабочих процессов. Применение предложенного подхода позволит повысить уровень автоматизации построения вычислительных рабочих процессов, генерировать рабочие процессы "на лету" основываясь на заранее подготовленном пуле функциональных сервисов и ограничений, наложенных онтологией предметной области.

**Ключевые слова:** *онтология, вычислительные рабочие процессы, функциональные сервисы, автоматизация рабочих процессов.*

### 1 Вступ

Розвиток інформаційних технологій спричиняє трансформацію всіх сфер життєдіяльності суспільства. Сучасне суспільство є неможливим без цифрової економіки, формування якої є в даний час одним з пріоритетних напрямків. Для широкого загалу "цифрова економіка" буде означати новий рівень цифрових сервісів, коли в онлайн переходять оплати комунальних

платежів, покупки і т.п., а для промисловості та бізнесу перехід в цифрову економіку отримав визначення в світі як Industry 4.0 - четверта індустріальна революція.

Сучасне бізнес середовище все більше надає значний обсяг послуг в цифровому просторі, тому телеком-оператори та провайдери онлайн-послуг мають гарантувати технологічну ефективність та інновації, постійно оновлюючи свої технології та послуги, адаптуючи та реконфігуруючи складні програмні системи, які функціонують в глобальному середовищі, надаючи послуги безперервно кінцевим користувачам.

Значну кількість робіт присвячено питанням автоматизації побудови робочих процесів, зокрема роботи [1, 2, 3, 4, 5]

Обчислювальні незалежні робочі процеси розробляють з використанням графічних стандартів, дозволяючи виконувати їх формалізацію з відображенням їх можливих потоків і переходів у схематичному вигляді. Аналіз показав, що на практиці обчислювальні незалежні робочі процеси зазвичай розробляються з використанням графічних позначень засобами, такими як BPMN 2.0, UML AD (UML Діаграма Дій), USLD та інструменти, такі як CA ERwin Process Modeler та Enterprise Architect [6].

Недоліки BPMN 2.0 чітко описані в [7]. Центральний аргумент проти використання регулярного BPMN полягає в тому, що керування ресурсами може бути виражено лише через певні абстракції (актори, ролі тощо) або виконання завдань в діалозі.

Тим не менш, BPMN, забезпечуючи можливість обчислювальної незалежної від обчислювальних робочих процесів трансформації (Business Process Execution Language (BPEL)), широко поширена в промисловості.

Короткий огляд методів та інструментів аналізу робочих процесів показав, що існують два типи аналізу, які враховують обчислювальний процес:

1. Аналіз часу проектування (моделювання та перевірка). Можуть бути використані засоби моделювання Монте-Карло, а також аналіз мереж Петрі, оскільки існують підходи трансформації для BPMN, UML AD, EPC, BPEL до мереж Петрі з їх подальшим аналізом. Діаграми USLD можуть бути проаналізовані за допомогою онтологічного аналізу послуг [8]

2. Аналіз часу виконання (наприклад, процес видобутку на основі журналів виконання) [9]

Для таких областей аналізу використовуються такі програмні засоби, як Pegasus, Cactus, ASKALON, GLUE і т.д.. Всі згадані та проаналізовані поточні можливості для цього етапу завдання дуже обмежені. Недоліки методів та інструментів аналізу робочих процесів чітко описані в [10]. Центральною критикою є те, що етап аналізу вимог застосовується в основному вручну.

В роботі [6] також розглянуто алгебру побудови робочих процесів, але це більш складний підхід, бо потребує детальної структуризації всіх даних та сервісів, встановлення повного переліку зв'язків, що утворює додаткову складність побудови алгебраїчної структури, особливо, коли система є територіально розподіленою та має значну постійно змінювану кількість сервісів та робочих процесів.

В роботі [11] запропоновано інструментарій для побудови та аналізу робочих процесів, але все ж таки запропонований підхід не достатньо автоматизує саме формування послідовності обчислень в залежності від умов та обмежень предметної області.

В роботі [12] представлена модель обчислювальних робочих процесів і приведений огляд різних обчислювальних архітектур, таких як обчислювальний кластер, Grid, Cloud Computing і SOA для їх побудови в розподіленому середовищі. Проведено порівняльний аналіз можливостей архітектур для реалізації обчислювальних робочих процесів, в ході якого визначено, що робочі процеси доцільно реалізовувати на основі SOA, оскільки вона відповідає всім вимогам до базової інфраструктури і забезпечує високу ступінь розподіленості обчислювальних вузлів, а також їх міграції та інтеграції з іншими системами в гетерогенному середовищі.

Таким чином, можна зробити висновок, що стратегії адаптації систем під постійні зміни вимог користувачів за рахунок автоматизованого проектування та реінжинірингу послуг (тобто проектування та модифікацій існуючих бізнес-процесів) погано формалізовано та перевірено, вимагається декілька ітерацій за участі аналітиків, системних архітекторів, значних витрат часу та ресурсів.

Таким чином, можна констатувати, що існує доволі багато інструментальних засобів для опису та побудови робочих процесів, але всі вони слабо автоматизовані та потребують безпосередньої участі людини. Рішень, які б дозволили виконувати автоматизований реінжиніринг робочих процесів у зв'язку зі зміною мета-описів функціональних сервісів, на даний момент не існує.

Підхід, запропонований у цьому дослідженні, може дозволити вирішити цю проблему завдяки використанню чітко визначених моделей робочого процесу, сервісів, предметної області, визначеного набору правил побудови зв'язків між функціональними сервісами у вигляді онтологічної моделі, яку легко корегувати, а також запропонованих інструментальних засобів побудови та модифікації бізнес процесів.

Ця стаття структурована наступним чином: в розділі 2 описується загальна постановка задачі дослідження. В розділі 3 приводиться опис моделі представлення сервісів для побудови обчислювальних робочих процесів. В розділі 4 надається формалізований опис інфраструктури підтримки виконання робочих процесів. В розділі 5 демонструється автоматизована побудова робочого процесу із застосуванням онтології.

## **2 Постановка задачі досліджень**

Онтологічна модель робочих процесів дозволяє будувати формальні описи процесів, поєднуючи функціональні сервіси зв'язками у вигляді обмежень предметної області, які відображатимуть взаємодію між ними, а онтологічна модель функціональних сервісів зберігатиме їх формальні мета-описи, з'являється можливість утворення деякого інтелектуального середовища, на основі якого можливо автоматично будувати робочі процеси в залежності від стану даних предметної області.

Перевагою онтологічного підходу є те, що онтологічна модель будується із застосуванням зв'язків у вигляді графів, дозволяючи застосовувати математичний апарат теорії графів для генерації робочих процесів в режимі реального часу, базуючись на описі вхідних та вихідних параметрів кожного функціонального сервісу [13].

Обчислювальним робочим процесом (workflow) будемо називати певний опис послідовності обчислювальних дій (етапів обробки даних), представлених у вигляді окремих структурних програмних компонент, що відповідають за конкретну функціональність і працюють автономно.

Модель обчислювальних робочих процесів служить для подання ланцюжка реальних обчислювальних задач, оцінки та аналізу їх ефективності, а також опису часто повторюваної послідовності операцій обробки даних.

Для відображення потоку робіт застосовують блок-схему або граф, який складається із операцій (функціональних сервісів), символів логіки, зв'язків. Послідовність виконання обчислень позначається стрілками.

Потік робіт з інформаційної точки зору – спосіб подання інформації до різних об'єктів, які приймають участь в робочому процесі.

Зазвичай робочий процес може бути описаний з використанням формальних або неформальних методів потокових діаграм, які показують спрямовані потоки між етапами обробки. Окремі кроки обробки або компоненти робочого процесу можуть бути в основному визначені трьома параметрами [14]:

- Опис входу: інформація, дані та ресурси, необхідні для робочого процесу.
- Правила перетворення: алгоритми, які можуть виконуватися людиною, програмою, чи обома.
- Опис виходу: інформація, дані та ресурси, оброблені робочим процесом та готові для передачі в сервіси для подальшої обробки.

Компоненти можуть бути поєднані лише за умови, якщо вихідна інформація одного компоненту відповідає вхідній інформації наступного компоненту, а також задовольняє умовам предметної області. Таким чином, семантичний опис має включати обов'язково опис вхідних, вихідних даних та мета-опис функціональних сервісів, що виконуються в даному робочому процесі. Якщо існує кілька методів (функціональних сервісів) обробки одних і тих же даних, то потрібно додати мета-опис алгоритмів, які виконують задану функцію, а також бажано вказати характеристики ефективності виконання заданих функцій, такі як: точність, швидкість, та інші.

Це пов'язано з тим, що інформація, якою оперує робочий процес, є слабко зв'язною різноструктурованою і може бути оброблена різними методами в залежності від характеристик потоку вхідних даних. Але з точки зору обробки даних, інформацію потрібно структурувати та навчити систему на конкретних даних для можливості більш ефективного використання різних методів.

Система керування робочим процесом – це програмна система, яка слугує для підготовки, виконання та моніторингу визначеного порядку робочих процесів та задач для підвищення продуктивності та зменшення витрат, потрібних для виконання робочого процесу.

### 3 Опис моделі представлення сервісів для побудови обчислювальних робочих процесів

В багатьох випадках одну й ту ж задачу можна вирішити різними методами. Характер обробки даних визначається імовірнісними властивостями спостережень, проведених на деяких даних, тому для опису всіх правил встановлення зв'язків на основі вхідних даних з метою визначення найбільш ефективного методу вирішення однієї і тієї ж задачі може бути використано онтологію.

Розглянемо запропонований підхід щодо автоматизації побудови онтології предметних областей, робочих процесів та всіх компонентів (функціональних сервісів, зв'язків та правил), за допомогою яких формується робочий процес.

Функціональний сервіс ( $S$ ) – сервіс, що приймає на вхід набір даних, виконує операцію над ними та віддає на вихід нові дані. Мета-описи функціональних сервісів знаходяться в базі даних. Всі мета-описи функціональних сервісів кожної предметної області зберігаються в окремих колекціях бази даних. Сервіси можуть бути простими та комплексними.

Розглянемо модель простого сервісу:  $S = \{Code, M, P\}$ , де:

$Code$  – програмний код функціонального сервісу

$M$  – мета-опис функціонального сервісу

$P$  – фізичний шлях до функціонального сервісу

Мета-опис функціонального сервісу ( $M$ ) – це опис функції, її вхідних та вихідних параметрів для подальшого відображення в користувацькому інтерфейсі та для врахування обмежень при побудові зв'язків у робочому процесі.

Мета-опис простого сервісу може бути представлено у вигляді:

$$M = \{ID, F, \{I_1, \dots, I_n\}, \{O_1, \dots, O_n\}, T\},$$

де:

$M$  – мета-опис, що включає такі параметри:

$ID$  – ідентифікатор мета-опису, наприклад:  $ObjectID("5cdd4fd0598ec622d023e048")$

$F = \{N, D, R, P\}$  – множина параметрів для опису функціонального сервісу;

$I_i = \{N_i, D_i, R_i, T_i\}$  – множина параметрів для опису вхідних параметрів;

$O_i = \{N_i, D_i, R_i, T_i\}$  – множина параметрів для опису вихідних параметрів, де:

$N$  – назва об'єкту (функціонального сервісу, вхідного/вихідного параметрів), наприклад «*checkBankAccount*»

$D$  – опис об'єкту; характеристика обчислювальних операцій функціонального сервісу, наприклад: «Сервіс перевіряє, чи існує дана картка в банку призначення»

$R$  – зручна для розуміння людини назва об'єкту «Перевірка наявності рахунку в банку отримувача»

$P$  – шлях до функціонального сервісу «./bank/checkBankAccount.js»

$T$  – тип сервісу: «function», «process»; тип змінної «number», «string» ...

$$Proc = \{CPU, R, t_{max}, \dots\}$$

де:

$Proc$  – опис нефункціональних параметрів сервісу;

$C$  – кількість обчислювальних ядер, що потрібні для виконання сервісу;

$R$  – кількість оперативної пам'яті;

$t_{max}$  – максимальний час для виконання сервісу;

... – інші параметри.

Мета-описи функціонального сервісу зберігаються в базі даних в такому вигляді:

```
{
  _id,
  function: {name, description, readable_name, path},
  input: [{name, description, readable_name, type}],
  output: [{name, description, readable_name, type}],
  proc: {cpu, ram, tmax, ...}
  type
}
```

Складні сервіси можуть бути представлені на рисунку нижче (див. Рис.1).

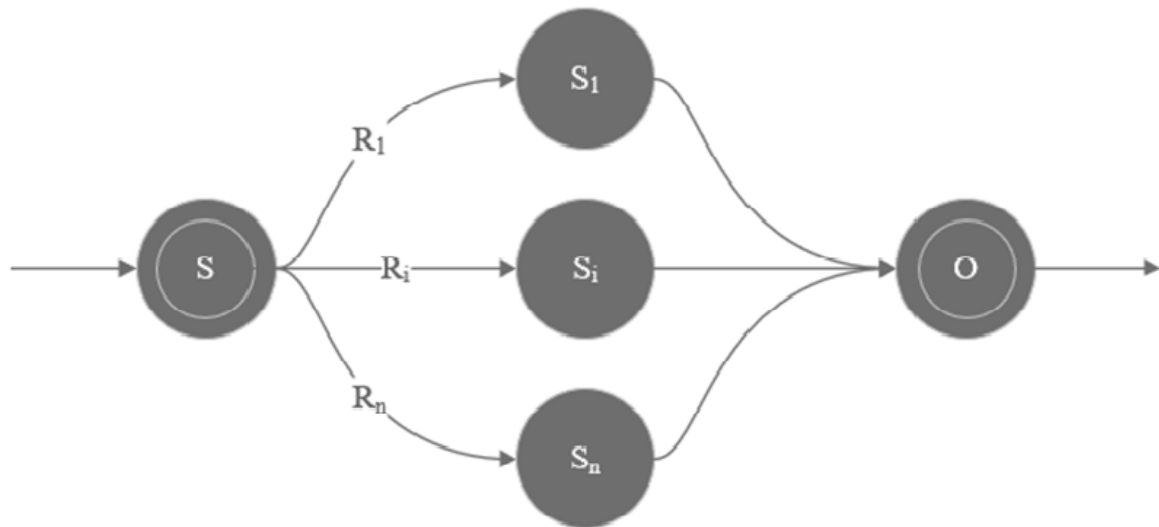


Рис.1 Приклад складного сервісу

В алгебраїчному вигляді складний сервіс може бути записано:

$S_{complex} = \{I, \{R_1, \dots, R_n\}, \{S_1, \dots, S_n\}, O, Proc\}$ , де:

$S_{complex}$  – набір параметрів, що описують комплексний сервіс

$I$  – вхідні дані сервісу

$R_i$  – правило вибору методу

$S_i = \{I \cap R_i\}$  – метод обробки даних або простий сервіс, що може бути вибрано при застосуванні

$R_i$  правила до вхідних параметрів

$O$  – вихідні дані сервісу

$Proc = \{CPU, R, t_{max}\}$  – набір нефункціональних параметрів, які набувають значення вибраного методу обробки

#### Формалізований опис робочих процесів

Онтологія робочих процесів складається із функціональних сервісів, та зв'язків між ними, що побудовані з врахуванням обмежень предметної області.

Предметна область – множина всіх об'єктів, класів та атрибутів, що пов'язані між собою логічними зв'язками.

Робочий процес – процес, що складається із одного або декількох функціональних сервісів, що пов'язані між собою предметною областю та логічними зв'язками.

Робочий процес представляється у такому вигляді:

$P = \{I, \{F_1, \dots, F_n\}, O, Proc\}$ , де:

$P$  – робочий процес, що включає такі параметри:

$I$  – опис вхідних параметрів

$F_i$  – використані функціональні сервіси

$O$  – опис вихідних параметрів

$Proc = \{CPU_{max}, R_{max}, t_{max}\}$  – нефункціональні параметри робочого процесу, де:

$CPU_{max}$  – максимально потрібна кількість обчислювальних ядер

$R_{max}$  – максимально потрібна кількість оперативної пам'яті

$t_{max}$  – максимально допустимий сумарний час виконання робочого процесу, що складається із суми  $t_{max}$  всіх функціональних сервісів.

Зв'язки між сервісами ( $L$ ) – логічні зв'язки, що вказують на порядок обробки даних, встановлюють порядок передачі параметрів із одного сервісу в інший, можуть бути встановлені лише за умови, що два сервіси є логічно зв'язними та хоча б один із вихідних параметрів першого сервісу відповідає хоча б одному вхідному параметру іншого сервісу.

#### 4 Формалізований опис інфраструктури підтримки виконання робочих процесів

В автоматизованих системах виконання робочих процесів виконується за допомогою систем керування робочими процесами та можуть бути описані за допомогою діаграм потоку даних.

Формально система керування робочими процесами може бути представлена у вигляді:

$$WfMS = \{ S_r, S_d, S_c, S_a, S_{ai} \},$$

де:  $S_r$  – система маршрутизації, виконує маршрутизацію потоку інформації або документообігу, передає інформацію від попереднього робочого елемента до наступного;

$S_d$  – система розподілу, виявляє виняткові обставини і передає інформацію призначеним компонентам системи;

$S_c$  – система координації, координує одночасну діяльність, запобігаючи конфліктам ресурсів (виділення процесорного часу та оперативної пам'яті) або конфліктам пріоритетів;

$S_a$  – система агентів, автоматично запускає сервіс, контролює його виконання та отримує від нього дані;

$S_{ai}$  – система помічників, здійснює регулювання виборів наступних методів та надає пропозиції, який з методів може бути використаний наступним.

Компоненти діаграми потоку даних може бути записано у вигляді:

$$DFD = \{ D_p, D_{df}, D_w, D_t \},$$

де:  $D_p$  – процес, частина системи, яка перетворює входи у виходи;

$D_{df}$  – потік даних, показує передачу інформації з однієї системи на іншу;

$D_w$  – склад даних, використовується для зберігання даних для подальшого використання;

$D_t$  – термінал доступу, інтерфейс взаємодії системи із зовнішнім середовищем.

### 5 Автоматизована побудова робочих процесів із застосуванням онтології

Для побудови робочого процесу потрібно виконати операцію перетину кожний з кожним для всіх сервісів за умови існування логічного зв'язку, а також зв'язку по даним між ними. При цьому вхідні дані визначатимуть той сервіс, з якого починаються обчислення за умови співпадіння даних з їх мета-описами в сервісі.

Формально побудова робочого процесу може бути представлено у вигляді:

$$P = \{ \forall M_{вик,i} \cap \forall M_{вик,j} \mid \exists L_{ij} \}, \text{ де:}$$

$P$  – згенерований робочий процес, що складається з:

$$M_{вик} = \{ M_{зад} \cap M_{обл} \} \text{ – вибір сервісів, що використані в робочому процесі,}$$

де:

$M_{вик}$  – функціональні сервіси, що використані в робочому процесі

$M_{зад}$  – задані користувачем функціональні сервіси

$M_{обл}$  – функціональні сервіси даної предметної області

$L_{ij} = \{ M_{вик,i} \cdot O = M_{вик,j} \cdot I \}$  – визначення зв'язків робочого процесу за правилом сумісних вхідних

та вихідних параметрів, де:

$L_{ij}$  – зв'язки робочого процесу

$M_{вик,i} \cdot O$  – вихідний параметр  $i$ -того функціонального сервісу;

$M_{вик,j} \cdot I$  – вхідний параметр  $j$ -того функціонального сервісу

$M_{обл} = \{ M_{всі} \in SA \}$  – вибір всіх сервісів, що належать предметній області,

де:

$M_{обл}$  – функціональні сервіси заданої предметної області

$M_{всі}$  – всі, доступні в репозиторії, функціональні сервіси

$SA$  – задана предметна область

Збереження робочих процесів забезпечено за допомогою мови опису онтологічних моделей – OWL. Кожен робочий процес зберігається в окремій онтології.

Приклад опису фрагменту онтологічної моделі засобами мови OWL наведено на рисунку нижче (див. Рис.2).

```
<?xml version="1.0"?>
<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://www.semanticweb.org/andriy2/ontologies/2019/4/name"
  ontologyIRI="http://www.semanticweb.org/andriy2/ontologies/2019/4/name">
  <Prefix IRI="http://www.semanticweb.org/andriy2/ontologies/2019/4/name"/>
  <Declaration>
    <Class IRI="#F1"/>
  </Declaration>
  <DisjointClasses>
    <Class IRI="#F1"/>
  </DisjointClasses>
</Ontology>
```

Рис.2 Фрагмент представлення онтологічної моделі засобами мови OWL

Даний метод дозволяє будувати робочі процеси, враховуючи всі обмеження предметної області та використовуючи мета-описи функціональних сервісів для встановлення зв'язків.

Використовуючи базу даних як джерело метаданих та онтологію предметних областей як правила для встановлення зв'язків між функціональними сервісами може бути побудовано робочі процеси та встановлено зв'язки між ними в автоматизованому вигляді. Такий підхід дозволяє пришвидшити створення нових процесів без детального знання мови опису онтології OWL та мови самих функціональних сервісів JavaScript. Робочий процес буде згенеровано автоматично завдяки застосуванню запропонованого підходу.

#### Реалізація прототипу робочого процесу на основі онтологій

Для проведення експерименту щодо побудови робочого процесу було використано:

- Інтегроване середовище розробки WebStorm;
- мову програмування JavaScript у вигляді серверного компоненту на Node.js для генерації онтології робочого процесу в автоматизованому вигляді
- базу даних MongoDB для збереження метаданих функціональних сервісів
- Protege для перегляду та редагування онтології

Фрагмент програмного коду для формування онтологічної моделі засобами JavaScript наведено на рисунку нижче (див. Рис.3).

```
let arr = await funs.find({
  'function.name': {
    $in: [
      'F1', 'F2', 'F3', 'F4', 'F5', 'F6'
    ]
  }
}).toArray();
arr.forEach(a => owl.addBundle(a));
```

Рис.3 Вибір функціональних сервісів з бази даних

Фрагмент програмного коду для формування зв'язків між онтологічними об'єктами предметної області, що визначаються вхідними та вихідними параметрами функціональних сервісів наведено на рисунку нижче (див. Рис.4):



```

arr.forEach(fun1 => {
  arr.forEach(fun2 => {
    fun1.output.forEach(var1 => {
      if (!vars_o[var1.name]) {
        vars_o[var1.name] = {f: fun1.function.name, v: false};
      }
      fun2.input.forEach(var2 => {
        if (!vars_i[var2.name]) {
          vars_i[var2.name] = {f: fun2.function.name, v: false};
        }
        if (var1.name === var2.name && var1.type === var2.type) {
          vars_o[var1.name].v = true;
          vars_i[var2.name].v = true;
          owl.addLink(fun1.function.name, 'output', var1.name);
          owl.addLink(var2.name, 'input', fun2.function.name);
        }
      }
    }
  })
});

```

Рис.4 Встановлення зв'язків між функціональними сервісами

Фрагмент згенерованого робочого процесу із застосуванням запропонованого підходу наведено на рисунку нижче (див. Рис.5)

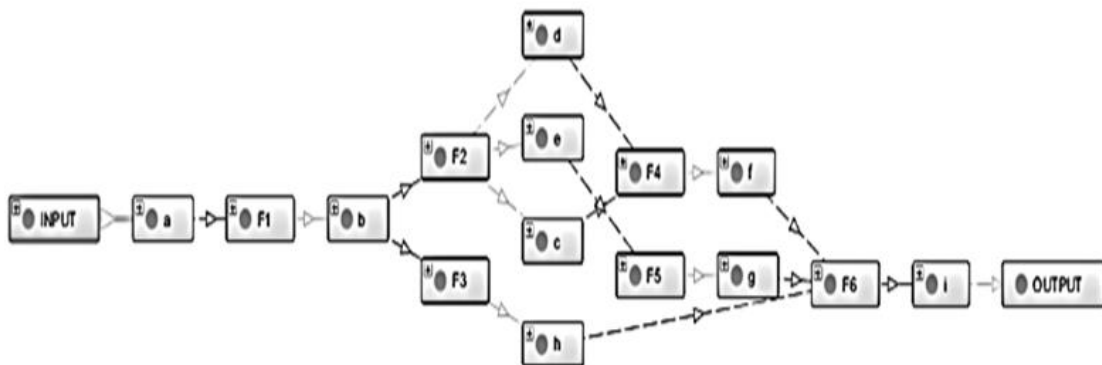


Рис.5 Автоматично згенерована онтологія робочого процесу

Таким чином запропонований підхід дозволяє автоматизувати формування обчислювального робочого процесу на основі функціональних сервісів та обмежень, накладених онтологією предметної області, а також дозволяє керувати процесом обчислень в залежності від вхідних даних.

## 6 Висновки

Проведений огляд технологій побудови обчислювальних робочих показав, що наявною є необхідність постійного реінженірингу бізнес-процесів в предметних областях, а відповідно підвищення рівня їх автоматизації. Разом з тим, ефективних рішень, які б дозволили виконувати автоматизований реінжиніринг обчислювальних робочих процесів, на даний момент не існує.

В роботі запропоновано формалізований опис онтологічних моделей предметної області, функціонального сервісу та робочих процесів, а також операції автоматизованого формування робочого процесу за допомогою зв'язків, що встановлюються між онтологічними моделями.

У статті представлений підхід до побудови обчислювальних робочих процесів, який заснований на чітко визначених онтологічних моделях предметної області, онтологічній моделі набору правил побудови зв'язків між функціональними сервісами, і онтологічній моделі самих робочих процесів, що динамічно генеруються. Проведено експериментальні дослідження даного

підходу та розроблено прототип інструментарію автоматизованої побудови обчислювальних робочих процесів.

Використання запропонованого підходу щодо побудови обчислювального робочого процесу із застосуванням онтології предметних областей та мета-описів функціональних сервісів дозволяє автоматизувати процес їх побудови, зокрема вибір методу (функціонального сервісу) обробки вхідних даних із множини аналогічних сервісів, який найбільше відповідає умовам зв'язків, що визначається цим потоком та є дуже важливим фактором в системах реального часу. В залежності від даних може бути вибраний найбільш ефективний метод для їх найшвидшої обробки.

Подальші дослідження будуть присвячені більш детальному розгляду процесу автоматизованої побудови робочих процесів із наборів функціональних сервісів, зокрема, автоматизованої генерації програмного коду виконання робочого процесу.

#### ЛІТЕРАТУРА

1. Workflow Patterns: веб-сайт. URL: <http://www.workflowpatterns.com>.
2. Globa L., Popova M., Yushko N. Improved Approach to Quality Control of Telecommunication Service Providers. Materials of the 8th International Conference on Applied Innovations in IT, (ICAИТ), March 2020. Keten, Germany: 2020. P. 29–34. URL: <https://opendata.uni-halle.de/handle/1981185920/32929>
3. Luntovskyy A., Globa L. Performance, Reliability and Scalability for IoT. 2019 International Conference on Information and Digital Technologies (IDT), 25-27 June 2019. Zilina, Slovakia: IEEE Digital Library, 2019. T.1. P. 316–322. DOI: 10.1109/DT.2019.8813679 URL: <https://ieeexplore.ieee.org/document/8813679>
4. Workflow Automation. Веб-сайт. URL: [https://web.archive.org/web/20130907014418/http://nocsmart.com/index.php?option=com\\_content&view=article&id=17&Itemid=135](https://web.archive.org/web/20130907014418/http://nocsmart.com/index.php?option=com_content&view=article&id=17&Itemid=135).
5. Новогрудська Р.Л. Інформаційна технологія створення та підтримки порталів інженерних знань : автореф. дис. на здобуття наук. ступеня канд. техн. наук : 05.13.06. Київ, 2015. 20 с.
6. Börger E. Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. In *Software and Systems Modeling*. Springer, 2011. P. 305–318.
7. Modern Business Process Automation: YAWL and its Support Environment / Adams M., ter Hofstede A.H.M., Russell N. Berlin, Germany: Springer-Verlag, 2010. 616 p.
8. Globa L., Kot T., Schill A., Strunk A. Method of IBIS design and workflow realization. "Polish J. of Environ. Stud " 2009. Vol. 18, № 4a. P. 35-38.
9. Globa L., Kot T., Schill A. Applying business process modeling method when Telecommunication services development. *СВЧ-техника и телекоммуникационные технологии*: 2011 год: материалы 121-й Международной Крымской конференции (КрыМиКо), 2011 г. Севастополь, Крым, Украина: IEEE Catalog Number СП 1788, 2011. T.1. С. 457 – 458. ISBN 978-966-335-352-4
10. Kot T., Reverchuk A., Globa L., Schill A. (2012): A novel approach to increase efficiency of OSS/BSS workflow planning and design. Materials of the 15th International Conference Business Information Systems, 21–23 May 2012. Vilnius; Lithuania: Springer, 2012. Vol. 117. P. 142-152. DOI: 10.1007/978-3-642-30359-3\_13
11. Abdukalykov R., Hussain I., Kassab M., Ormandjieva O. Quantifying the Impact of Different Non-functional Requirements and Problem Domains on Software Effort Estimation. 2011. Ninth International Conference on Software Engineering Research, Management and Applications: materials of the 2011 Ninth International Conference, 10-12 Aug. 2011, Baltimore, MD, USA, 2011. DOI: 10.1109/SERA.2011.45. URL: <https://ieeexplore.ieee.org/document/6065634>
12. Гаевой В.В., Глоба Л.С. Анализ архитектур для построения вычислительных рабочих процессов в распределённой среде . Вісник Харківського національного університету імені В.Н. Каразіна, серія «Математичне моделювання. Інформаційні технології. Автоматизовані системи управління». 2020 Вип. 46. С. 7-16.
13. Globa L., Novograduska R., Koval A., Senchenko V. Ontology for Application Development. *Ontology in Information Science* / за ред. Ciza Thomas. IntechOpen, 2018. P. 29 – 53

DOI: 10.5772/intechopen.74042 URL: <https://www.intechopen.com/books/ontology-in-information-science/ontology-for-application-development>

14. Business process execution language for Web services: веб-сайт. URL: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>.

#### REFERENCES

1. Workflow Patterns Resource. [Online]. Available: <http://www.workflowpatterns.com/>
2. L. Globa, M. Popova, N. Yushko, "Improved Approach to Quality Control of Telecommunication Service Providers", in *Proc. 8th International Conference on Applied Innovations in IT, (ICAIIIT)*, Keten, Germany, 2020, pp. 29-34, URL: <https://opendata.uni-halle.de/handle/1981185920/32929>
3. A. Luntovskyy, L. Globa, "Performance, Reliability and Scalability for IoT", in *Proc. 2019 International Conference on Information and Digital Technologies (IDT)*, Zilina, Slovakia, 2019, pp.316-322, DOI: 10.1109/DT.2019.8813679, URL:<https://ieeexplore.ieee.org/document/8813679>
4. Workflow Automation. [Online]. Available: [https://web.archive.org/web/20130907014418/http://nocsmart.com/index.php?option=com\\_content&view=article&id=17&Itemid=135](https://web.archive.org/web/20130907014418/http://nocsmart.com/index.php?option=com_content&view=article&id=17&Itemid=135).
5. R. Novogradskaya. "Information technology for creating and maintaining engineering knowledge portals": PhD thesis, Igor Sikorsky Kyiv Polytechnic Institute, Kiev, 2015. [in Ukrainian]
6. E. Börger. "Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL", in *Software and Systems Modeling*, Börger E., Eds. Springer-Verlag 2011, pp. 305–318. DOI 10.1007/s10270-011-0214-z
7. M. Adams, A.H.M. ter Hofstede, N. Russell. *Modern Business Process Automation: YAWL and its Support Environment*. Berlin, Germany: Springer-Verlag, 2010.
8. L. Globa, T. Kot, A. Schill, A. Strunk. "Method of IBIS design and workflow realization", *"Polish J. of Environ. Stud "*, Vol. 18, no. 4a, pp. 35-38, 2009.
9. L. Globa, T. Kot, A. Schill, "Applying business process modeling method when Telecommunication services development", in *Proc. 21st International Crimean Conference "Microwave and Telecommunication Technology" (CriMiCo'2011)*, Sevastopol, Crimea, Ukraine, 2011, pp. 457 - 458. ISBN 978-966-335-352-4 URL: <https://ieeexplore.ieee.org/document/6069007>
10. Kot, T., Reverchuk, A., Globa, L., Schill, A., "A novel approach to increase efficiency of OSS/BSS workflow planning and design", in *Proc. 15th International Conference Business Information Systems*, Vilnius; Lithuania, 2012, pp. 142-152.
11. R. Abdukalykov, I. Hussain, M. Kassab, O. Ormandjieva, "Quantifying the Impact of Different Non-functional Requirements and Problem Domains on Software Effort Estimation", in *Proc. 2011 Ninth International Conference on Software Engineering Research, Management and Applications*, Baltimore, MD, USA, 2011. DOI: 10.1109/SERA.2011.45. URL: <https://ieeexplore.ieee.org/document/6065634>
12. V. Gaievyi, L. Globa, "Analysis of architectures for building computational workflows in a distributed environment" , *Bulletin of V.N. Karazin Kharkiv National University, series «Mathematical modeling. Information technology. Automated systems»*, Vol. 46, pp.7-16, 2020. [in Russian].
13. L. Globa, R. Novogradskaya, A. Koval, V. Senchenko, *Ontology for Application Development*, Ciza Thomas, Eds. IntechOpen, 2018, pp. 29 – 53, DOI: 10.5772/intechopen.74042, 2018, URL: <https://www.intechopen.com/books/ontology-in-information-science/ontology-for-application-development>
14. Business process execution language for Web services. [Online]. Available: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>.