

DOI: <https://doi.org/10.15276/ict.03.2026.06>

УДК 004.942

Шаблонізована мова перетворень результатів хмарних обчислень для мовонезалежної візуалізації

Ілуща Андрій Сергійович¹⁾

ORCID: <https://orcid.org/0009-0007-3445-8481>; ilutsa.a.s@op.edu.ua. Scopus Author ID: 59735462200

Павленко Віталій Данилович¹⁾

ORCID: <https://orcid.org/0000-0002-5655-4171>; pavlenko_vitalij@ukr.net. Scopus Author ID: 54401442600

¹⁾ Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Метою роботи є розробка спеціалізованої мови шаблонів перетворень результатів хмарних обчислень, яка забезпечує їх структурування та візуалізацію незалежно від мови програмування інструментальних засобів. Необхідність такого рішення зумовлена тим, що у традиційних підходах обчислення, аналіз і графічне подання результатів часто реалізуються в одному програмному коді, тому зміна форми подання потребує його редагування. Запропоновано відокремити обчислювальну процедуру від візуалізації результатів. Мову подано як формалізовану систему синтаксичних конструкцій і правил їх інтерпретації транслятором. Синтаксис охоплює керівні рядки, блоки візуалізації, серії даних і службові символи для опису графічного та табличного подання. Процес трансляції включає перевірку структури і змісту шаблону, формування внутрішнього ієрархічного представлення та підготовку даних для модулів візуалізації. Внутрішнє представлення подано як структуру, подібну до абстрактного синтаксичного дерева, у якій блоки візуалізації, серії даних і параметри відображення мають формалізоване представлення. Розроблена мова забезпечує перетворення результатів у стандартизовані структури, уніфікує їх представлення та спрощує подальший аналіз у web-орієнтованому хмарному середовищі. Практичним контекстом застосування розробленої мови є нейрофізіологічні дослідження на основі даних айтрекінгу, в яких результати обробки та моделювання потребують подальшого структурованого представлення і візуалізації.

Ключові слова: хмарні обчислення; web-орієнтовані програмні засоби; спеціалізована мова; мова шаблонів перетворень; транслятор; трансляція; візуалізація даних; айтрекінг

Для цитування: Ілуща А. С., Павленко В. Д. «Шаблонізована мова перетворень результатів хмарних обчислень для мовонезалежної візуалізації». *Інформатика. Культура. Техніка*. 2026; Том 3 № 1 (3): 75–84. DOI: <https://doi.org/10.15276/ict.03.2026.06>

Вступ. Сучасні дослідження дедалі частіше поєднують роботу з експериментальними даними, програмним кодом, обчислювальними алгоритмами та результатами їх виконання. У межах одного дослідження дані можуть проходити декілька послідовних етапів: отримання і зберігання, попередню обробку, математичне або комп'ютерне моделювання, виконання обчислень, аналіз та візуалізацію результатів. Використання хмарних технологій спрощує доступ до обчислювальних ресурсів і програмних засобів, однак саме по собі не усуває необхідність узгоджувати ці етапи між собою. Різні хмарні сервіси надають користувачу різний рівень контролю над програмним середовищем, а переносимість обчислювальних рішень між середовищами залишається окремою задачею.

На практиці дослідницький процес часто формується з декількох програмних засобів. Один інструмент використовується для підготовки даних, інший – для виконання алгоритмів, ще один – для побудови графіків або таблиць. За такого підходу зв'язок між даними, параметрами, програмним кодом і результатами частково підтримується самим дослідником. Додатковою проблемою є те, що спосіб представлення результатів часто задається безпосередньо у програмному коді конкретного проекту, тому зміна типу графіка, складу серій, підписів осей або табличної форми може потребувати редагування алгоритмічної частини або створення окремої процедури візуалізації.

Для наукових обчислень важливим є не лише отримання кінцевого результату, але й можливість повторити дослідження за тими самими даними та параметрами, порівняти декілька запусків і повторно використати вже підготовлені процедури, тому сталою є потреба забезпечувати зв'язок між даними, кодом, параметрами і результатами.

© Ілуща А., Павленко В., 2026

Стаття опублікована у відкритому доступі відповідно до умов ліцензії CC BY (<https://creativecommons.org/licenses/by/4.0/deed.uk>)

У цьому ланцюгу представлення результатів також має бути узгодженим: однакові за структурою результати важливо подавати за однаковими правилами незалежно від того, якою мовою програмування реалізовано обчислювальний алгоритм.

Таким чином, проблема полягає не лише в організації віддаленого виконання обчислень. Для цілісного дослідницького процесу необхідно відокремити логіку отримання результату від логіки його структуризації та візуалізації і водночас забезпечити між ними однозначний зв'язок. Такий підхід дозволяє змінювати спосіб представлення без зміни самого алгоритму, повторно використовувати однакові правила для різних проєктів і зменшувати кількість ручних дій, пов'язаних з підготовкою результатів до подальшого аналізу.

Аналіз останніх досліджень і публікацій. Сучасні дослідження хмарних технологій розглядають сервіси PaaS (Platform as a Service) і SaaS (Software as a Service) з позицій рівня абстракції, доступних обчислювальних ресурсів, масштабованості та можливості перенесення програмних рішень між хмарними середовищами [1], [2]. Для наукової роботи ці властивості є важливими, оскільки визначають спосіб виконання алгоритмів і доступу до даних. Водночас питання представлення отриманих результатів залишається переважно на рівні конкретного програмного засобу або бібліотеки, що використовується в проєкті.

Окремий напрям досліджень пов'язаний із забезпеченням відтворюваності наукових обчислень, де розглядаються принципи організації дослідницького програмного забезпечення, опису даних і зв'язку між складовими обчислювального процесу [3], [4], [5], [6]. Спільним для цих підходів є прагнення зберегти умови отримання результату та забезпечити можливість повторного виконання дослідження. З цього випливає необхідність фіксувати не тільки дані й параметри обчислення, але й правила, за якими отриманий результат перетворюється у форму, придатну для перегляду та аналізу.

Поширеними засобами організації наукових обчислень є Project Jupyter і Google Colab, які поєднують програмний код, текстові пояснення та результати виконання в межах інтерактивних документів [7], [8]. Такий формат є зручним для проведення обчислень і безпосереднього відображення результатів, однак спосіб побудови графіків і таблиць визначається програмним кодом, з прив'язкою до мови програмування, та використаними бібліотеками. Дослідження подібних рішень вказують на проблеми, пов'язані з порядком виконання, неявними залежностями, повторним використанням коду та відтворюваністю результатів [9], [10], [11], [12]. Перехід до хмарних обчислень сам по собі не усуває залежність способу представлення результатів від конкретної реалізації обчислювального алгоритму. Результати можуть формуватися у різних структурах і форматах залежно від мови програмування та програмних засобів, що використовуються для їх отримання. Унаслідок цього побудова графіків і таблиць потребує окремої обробки та налаштування для різних реалізацій.

Отже, наявні підходи добре вирішують окремі частини дослідницького процесу: надають обчислювальне середовище, поєднують код і результати, підтримують повторне виконання або допомагають контролювати зв'язки між етапами. Проте для дослідницьких задач, у яких один і той самий спосіб представлення має використовуватися для різних алгоритмів, що реалізовані на різних мовах програмування, залишається потреба у окремому формалізованому описі структури результату. Такий опис має бути незалежним від мови програмування обчислювального алгоритму і придатним для автоматичної інтерпретації у web-орієнтованому середовищі.

Нейрофізіологічні дослідження на основі даних айтрекінгу є одним із прикладів задач, для яких така вимога є практично значущою. Результати можуть включати часові залежності, декілька серій експериментальних даних, характеристики математичних моделей, таблиці показників і результати класифікації. Сучасні дослідження розглядають параметри рухів очей та інші цифрові показники як джерело даних для оцінювання неврологічного стану людини [13], [14], [15], [16]. Однак потреба у відокремленні обчислювального алгоритму від способу представлення його результатів не є специфічною лише для нейрофізіології та

виникає у широкому класі досліджень, де використовуються різні методи обробки даних і моделювання.

Наявна потреба у формалізації способу перетворення результатів обчислень у стандартизоване графічне або табличне представлення без прив'язки до мови програмування, у якій реалізовано обчислювальний алгоритм. Для цього необхідний окремий опис структури результатів і правил їх інтерпретації, який може бути використаний різними інструментальними засобами та повторно застосований у різних дослідницьких задачах.

Метою роботи є підтримка структуризації та візуалізації результатів хмарних обчислень незалежно від мови програмування інструментальних засобів шляхом розроблення мови шаблонів перетворень, яка задає формалізовані правила опису, інтерпретації та представлення одержаних результатів.

Місце мови шаблонів у процесі хмарних обчислень. Розроблено web-орієнтовані програмні засоби, що поєднують роботу з програмним кодом на рівні PaaS (Platform as a Service) і використання готових графічних інтерфейсів користувача на рівні SaaS (Software as a Service), підтримують серверне виконання коду, конструктор інтерфейсів для керування параметрами та централізоване зберігання результатів [17], [18], [19]. Такі засоби застосовуються для виконання дослідницьких задач, зокрема обробки даних айттрекінгу та оцінювання психофізіологічного стану людини [20], де результати обчислень можуть формуватися різними алгоритмами та потребувати подальшого представлення у вигляді графіків і таблиць. Саме на етапі представлення результатів використовується спеціалізована мова шаблонів, яка відокремлює опис їх структури та візуалізації від програмної реалізації обчислювального алгоритму.

У межах розроблених програмних засобів організації хмарних обчислень важливо відокремити отримання результатів від їх аналізу та представлення. У традиційних підходах ці етапи переважно реалізуються в програмному коді засобами самої мови програмування. Це прив'язує представлення результатів до конкретної мови програмування, ускладнює зміну способу аналізу та обмежує повторне використання алгоритмів.

У межах обчислювального процесу формування первинного результату та його представлення розглядаються як два послідовні, але відокремлені етапи. На першому етапі алгоритм обробляє вхідні дані відповідно до заданих параметрів. На другому етапі отриманий результат перетворюється у форму, придатну для аналізу користувачем. Таке розмежування визначає межу між алгоритмічною та інтерфейсною складовими і є основою для введення спеціалізованої мови шаблонів.

На етапі виконання алгоритму первинний результат визначається вхідними даними X , множиною параметрів P та алгоритмічною процедурою F :

$$Y = F(X, P), \quad (1)$$

де X – набір експериментальних даних; P – параметри обробки або моделювання; F – алгоритмічна процедура; Y – первинний результат обчислення.

У межах серії досліджень один і той самий алгоритм може виконуватися для декількох наборів параметрів:

$$Y_i = F(X, P_i), \quad i = 1, \dots, n, \quad (2)$$

що дозволяє отримувати порівнювані результати без зміни програмної реалізації алгоритму. Після завершення обчислень первинний результат має бути переведений у форму, придатну для аналізу користувачем.

Для цього вводиться окреме перетворення:

$$Y^* = T(Y). \quad (3)$$

Функція T визначає правила структуризації та подання результату.

Множина допустимих способів інтерпретації утворює клас шаблонів:

$$\mathcal{T} = \{T_1, T_2, \dots, T_n\}, \quad (4)$$

де кожен шаблон відповідає окремому способу представлення результатів: табличному, графічному або іншому структурованому поданню.

Таким чином, спеціалізована мова шаблонів визначає формальний простір перетворень, у якому один первинний результат може бути представлений у декількох формах без зміни алгоритмічної складової.

У загальному обчислювальному процесі мова шаблонів перетворень утворює окремий рівень між отриманням результатів та їх представленням (Рис. 1). Вона задає єдині правила опису вихідних даних і забезпечує їх узгоджену інтерпретацію незалежно від реалізації алгоритмів або мови програмування.

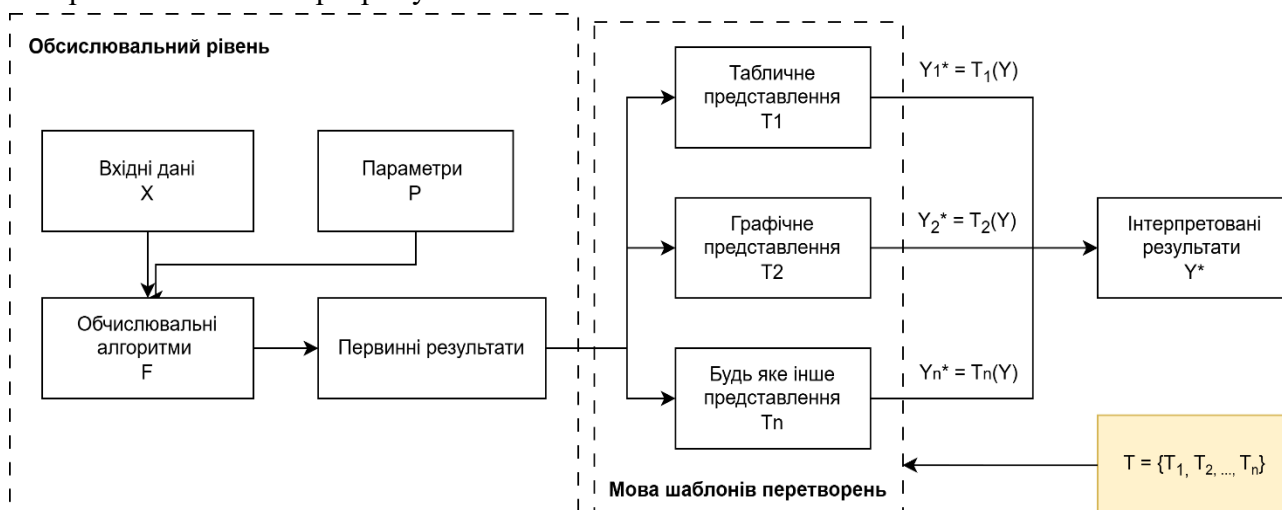


Рис. 1. Місце мови шаблонів перетворень у обчислювальному процесі

Джерело: розроблено авторами

Ключовою властивістю мови є декларативний характер. Замість опису алгоритму побудови графіка або таблиці користувач задає структуру результату та формат його подання. Обчислювальна процедура залишається відповідальною лише за формування первинних даних, а транслятор інтерпретує декларативний опис та передає уніфіковані структури відповідним модулям візуалізації. Завдяки цьому один спосіб подання може застосовуватися до результатів, сформованих різними мовами програмування.

У контексті нейрофізіологічних досліджень на основі даних айтрекінгу такий підхід дозволяє формалізувати типові процедури подання результатів: відображення часових залежностей, порівняння декількох серій експериментів, представлення перехідних характеристик моделей, формування таблиць діагностичних ознак та відображення результатів класифікації. Зміна типу подання не потребує зміни обчислювального алгоритму, що підтримує повторне використання дослідницьких сценаріїв.

Синтаксис спеціалізованої мови шаблонів перетворень. Синтаксис мови побудований на використанні текстових директив, які інтерпретуються розробленим транслятором незалежно від мови програмування, на якій написано інструментальні засоби. Основною одиницею є блок візуалізації, що описує одну структурну одиницю (таблицю, графік або інший тип представлення).

Початок блоку задається керівним рядком виду:

```
###visualizationType-Xaxis-Yaxis##Block Title##
```

У цій конструкції visualizationType визначає тип представлення результатів, зокрема line, table, column або scatter; Xaxis-Yaxis задає підписи вісей для графічного подання; Block Title визначає назву блоку. Такий запис у компактній формі задає контекст візуалізації і відокремлює його від безпосередніх значень даних.

Дані у межах блоку подаються у вигляді інформаційних рядків:

```
vector_name$data
```

Символ «\$» розділяє ідентифікатор набору даних та його значення. Ідентифікатор використовується як легенда графіка або заголовок стовпця таблиці, а data містить послідовність даних (array). Наприклад:

```
y1$array_values  
y2$array_values
```

Перший запис визначає одну послідовність значень, другий – іншу. Уніфікація текстового представлення дозволяє інтерпретувати дані однаково незалежно від того, яким способом і якою мовою програмування вони були сформовані.

Окремі серії даних у межах одного блоку розділяються службовим символом «#». Початок нового незалежного блоку задається конструкцією «###», що дозволяє описувати декілька візуалізацій (структурних одиниць) у межах одного результату обчислювального експерименту.

Додатково передбачено керування стилем відображення безпосередньо у заголовку серії:

```
vector_name-line_type-marker$
```

Параметр line_type визначає тип лінії, а marker – тип маркера. За відсутності цих параметрів використовуються стандартні значення. Такий підхід дозволяє задавати характеристики візуалізації без зміни програмної частини алгоритму.

Узагальнений опис багатосерійного графіка має вигляд:

```
###Line-Frames(time)-values##Model response##  
y1$array_values  
#  
y2$array_values  
#  
y3$array_values
```

Аналогічно формується табличне представлення:

```
###table##  
Feature$array_values  
#  
Precision$array_values  
#  
Recall$array_values
```

У наведених прикладах перший рядок визначає тип візуалізації та її параметри, а наступні рядки задають структуру даних. Отже, текстовий опис містить відомості, необхідні транслятору для формування відповідного подання. В одному результаті можуть послідовно розміщуватися декілька незалежних блоків, що дає змогу сформувати кілька графіків або таблиць у межах одного обчислювального експерименту.

Опис способу подання не залежить від API конкретної бібліотеки візуалізації і не потребує окремого програмування всіх етапів побудови графіка або таблиці. Запропонована мова створена саме для перетворення результатів обчислень у структури, які безпосередньо використовуються для візуалізації.

Формалізація процесу трансляції. Текстовий опис шаблону не може бути безпосередньо використаний модулем візуалізації. Його обробляє транслятор, який перевіряє шаблон, однозначно тлумачить його конструкції та формує внутрішні структури для побудови графіків і таблиць.

Процес трансляції складається з послідовних етапів: лексико-синтаксичного аналізу, семантичного аналізу, формування внутрішнього представлення та підготовки структур для візуалізації (Рис. 2).

На етапі синтаксичного аналізу перевіряється відповідність вхідного опису правилам мови. Контролюється коректність використання службових символів «###», «##», «\$» і «#», правильність структури блоків, а також наявність необхідних елементів: типу візуалізації, заголовка та даних. Результатом є структуроване представлення шаблону, придатне для подальшої перевірки.

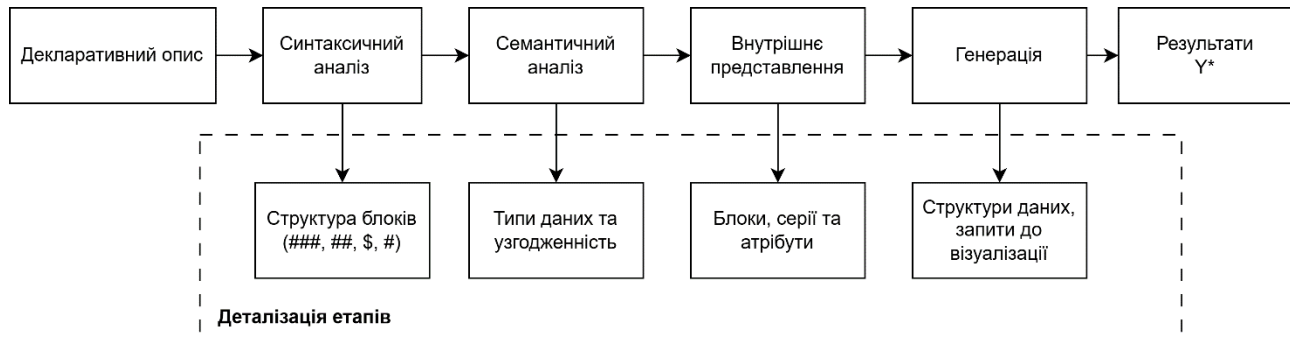


Рис. 2. Процес трансляції мови шаблонів перетворень

Джерело: розроблено авторами

Семантичний аналіз спрямований на перевірку змістовної коректності опису. На цьому рівні визначається узгодженість типів даних, відповідність структури даних вибраному типу візуалізації, коректність назв осей і кількості серій, а також наявність логічних суперечностей. Виконання цієї перевірки до етапу генерації дозволяє виявити помилки до моменту формування кінцевого представлення.

На наступному етапі формується внутрішнє представлення шаблону – уніфікована структура, яка відображає елементи декларативного опису у вигляді формалізованих об'єктів. Кожен блок візуалізації відповідає окремому об'єкту, серії даних подаються як структуровані набори, а параметри відображення зберігаються у вигляді атрибутів. Внутрішнє представлення не залежить від вихідного текстового формату та конкретної програмної реалізації.

Таке представлення доцільно розглядати як ієрархічну структуру, подібну до абстрактного синтаксичного дерева (Рис. 3). Вузол верхнього рівня відповідає блоку візуалізації та містить тип подання, заголовок, назви осей і множину дочірніх елементів. Вузли нижчого рівня відповідають серіям даних і містять ідентифікатор, набір значень та параметри стилю.

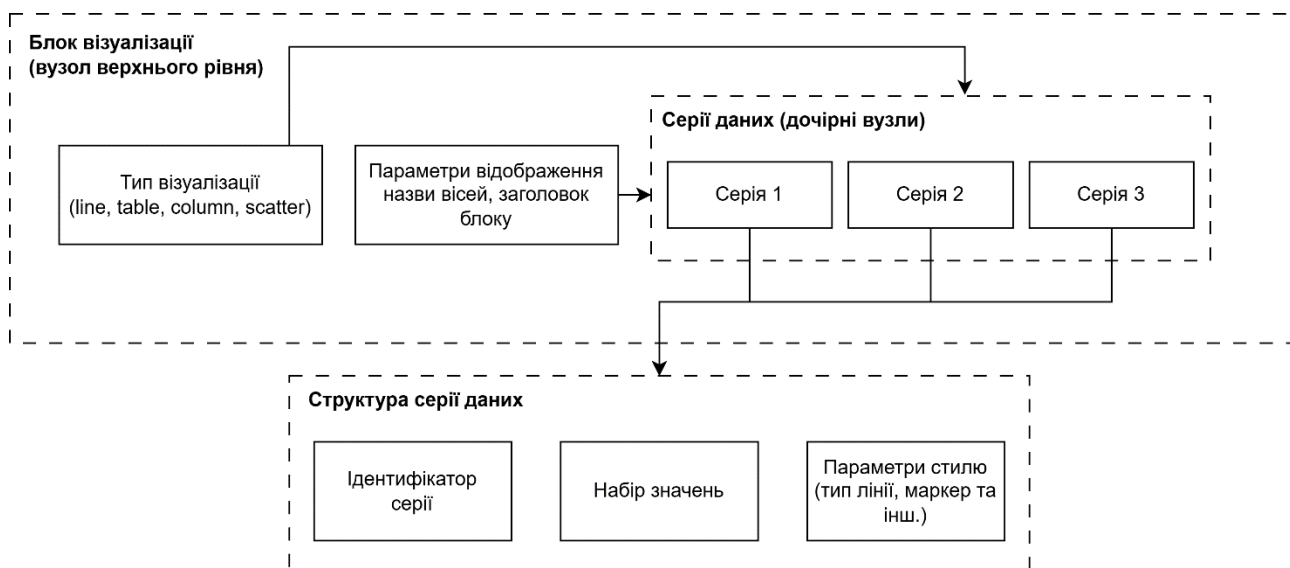


Рис. 3. Ієрархічна структура внутрішнього представлення шаблону.

Джерело: розроблено авторами

Ієрархічне представлення дозволяє надалі працювати з шаблоном незалежно від його початкового текстового запису. На завершальному етапі формуються дані для побудови графіків і таблиць та їх відображення у загальному web-інтерфейсі. Тип вузла верхнього рівня визначає вид представлення, а дочірні вузли містять дані та параметри їх відображення.

Важливою властивістю транслятора є незалежність від реалізації обчислювальних алгоритмів. Він працює з результатами їх виконання і декларативним описом подання, тому заміна мови програмування для створення інструментального засобу не потребує зміни логіки візуалізації за умови формування результату відповідно до визначених синтаксичних конструкцій. Це забезпечує модульність і мовонезалежність рівня представлення.

Програмні засоби та використання у дослідницькому процесі. Спеціалізована мова шаблонів реалізована у складі web-орієнтованих програмних засобів, що поєднують інтерфейсний (SaaS) і програмний (PaaS) рівні роботи з обчислювальними проєктами. Користувач може працювати з готовим інтерфейсом або редагувати програмний код, а результати після серверного виконання передаються до модуля трансляції та відображаються у браузері.

У загальному циклі взаємодії вхідні дані вибираються з централізованого сховища або передаються із зовнішніх джерел. Параметри обробки задаються через інтерфейс і автоматично передаються до програмного коду. Після виконання алгоритму результат містить текстовий опис відповідно до синтаксису мови шаблонів. Транслятор виділяє блоки візуалізації, формує внутрішню структуру та передає її відповідному модулю представлення. Взаємодію основних компонентів розроблених програмних засобів наведено на рисунку 4.

Технологія айтрекінгу використовується для отримання експериментальних даних, на основі яких виконуються попередня обробка, ідентифікація окуло-моторної системи, формування діагностичних ознак та оцінювання психофізіологічного стану. У цьому циклі спеціалізована мова застосовується після виконання обчислювального алгоритму і визначає, як первинні результати мають бути структуровані та відображені у web-інтерфейсі.

Для результатів моделювання можуть формуватися графіки перехідних характеристик і порівняльні серії для різних експериментальних умов. Результати формування ознак або класифікації можуть подаватися у табличному вигляді. Тип представлення, назви осей, заголовки, склад серій і параметри їх відображення задаються у шаблоні, тоді як алгоритмічна процедура залишається незмінною.

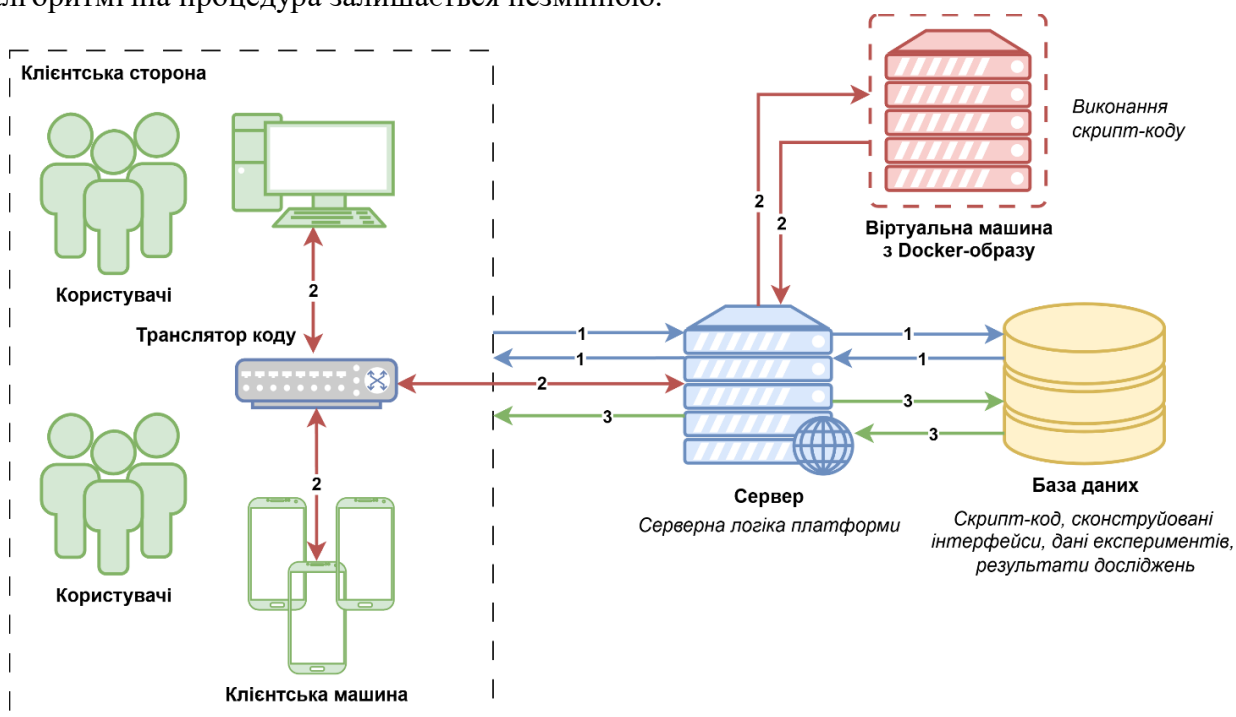


Рис. 4. Взаємодія основних компонентів розроблених програмних засобів

Джерело: розроблено авторами

У традиційному сценарії отриманий результат необхідно передати до окремого засобу візуалізації, налаштувати графік або таблицю та виконати їх побудову. У розроблених програмних засобах ці дії включено до загального циклу обробки: після серверного виконання транслятор аналізує керівні рядки та інформаційні блоки, формує внутрішню структуру і передає її відповідному компоненту представлення. У результаті графік або таблиця формуються без перенесення даних до окремого засобу, а дослідник може зосередитися на аналізі отриманих результатів.

Практичне значення мовнезалежного підходу полягає у використанні одного синтаксису для результатів алгоритмів, реалізованих мовами Python, JavaScript або іншими. За умови формування вихідних даних, відповідно до визначених синтаксичних конструкцій, той самий шаблон може застосовуватися до різних алгоритмів. Це відокремлює зміну форми представлення від зміни обчислювального коду, уніфікує підготовку графіків і таблиць та спрощує подальший аналіз результатів.

Висновки. Розроблено спеціалізовану мову шаблонів перетворень, що відокремлює формування первинного результату від його структуризації та представлення. Мову представлено як формалізовану систему синтаксичних конструкцій і правил їх інтерпретації транслятором. Визначено керівний рядок блоку, ідентифікатори серій, розділювачі даних, службові символи для формування декількох серій і незалежних блоків, а також параметри їх відображення. Такий опис дозволяє змінювати структуру подання без редагування обчислювального алгоритму за умови формування вихідних даних відповідно до визначеного синтаксису.

Формалізовано процес трансляції від текстового опису шаблону до внутрішніх структур, які використовуються для побудови графіків і таблиць. Процес охоплює перевірку синтаксису і змістовної узгодженості, формування внутрішньої ієрархічної структури та підготовку даних для модулів візуалізації. Внутрішнє представлення подано у вигляді структури, подібної до абстрактного синтаксичного дерева, де блоки візуалізації, серії даних і параметри відображення мають формалізоване представлення.

Інтеграція мови шаблонів у web-орієнтовані програмні засоби забезпечує використання одного синтаксису для візуалізації результатів алгоритмів, реалізованих мовами Python і JavaScript. У нейрофізіологічних дослідженнях на основі даних айтрекінгу це дозволяє включити формування графіків і таблиць до загального обчислювального процесу без перенесення логіки візуалізації до окремих модулів або інструментальних засобів. Уніфіковане представлення результатів і автоматизоване формування візуалізацій спрощують їх підготовку до подальшого аналізу.

Конфлікт інтересів: Автори декларують, що не мають конфлікту інтересів, зокрема фінансового, особистого, авторського чи будь-якого іншого характеру, який міг би вплинути на дослідження, а також на результати, опубліковані в цій статті

Фінансування: Дослідження проводилося без фінансової підтримки

Доступність даних: Усі дані доступні в цифровій або графічній формі в основному тексті рукопису

Використання засобів штучного інтелекту (ШІ): Автори підтверджують, що не застосовували інструментальні засоби ШІ для написання цієї роботи

СПИСОК ЛІТЕРАТУРИ

1. Younis, R., Iqbal, M., Munir, K., Javed, M. A., Haris, M. & Alahmari, S. A. “A comprehensive analysis of cloud service models: IaaS, PaaS, and SaaS in the context of emerging technologies and trend”. *International Conference on Electrical, Communication and Computer Engineering (ICECCE)*. 2024. p. 1–6. DOI: <https://doi.org/10.1109/ICECCE63537.2024.10823401>.
2. Moravcik, M., Segec, P., Kontsek, M. & Zidekova, L. “Model-Driven approach to cloud-portability issue”. *Applied Sciences*. 2024; 14 (20): 1–21. DOI: <https://doi.org/10.3390/app14209298>.
3. Lamprecht, A.-L. & Garcia, L. “Towards FAIR principles for research software”. *Data Science*. 2020; 3 (1): 37–59. DOI: <https://doi.org/10.3233/DS-190026>.

4. Soiland-Reyes, S., Sefton, P., et al. “Packaging research artefacts with RO-Crate”. *Data Science*. 2022; 5 (2): 97–138. DOI: <https://doi.org/10.3233/DS-210053>.
5. Leipzig, J., Nüst, D., Hoyt, C., Ram, K. & Greenberg, J. “The role of metadata in reproducible computational research”. *Patterns*. 2021; 2 (9): 1–28. DOI: <https://doi.org/10.1016/j.patter.2021.100322>.
6. Ahmad, Z., Jehangiri, A. I., Ala'anzy, M. A., Othman, M., Latip, R., Zaman, S. K. U. & Umar, A. I. “Scientific workflows management and scheduling in cloud computing: taxonomy, prospects, and challenges”. *IEEE Access*. 2021; 9: 53491–53508. DOI: <https://doi.org/10.1109/ACCESS.2021.3070785>.
7. Siddik, M. S., Li, H. & Bezemer, C.-P. “A systematic literature review of software engineering research on Jupyter Notebook”. *Journal of Systems and Software*. 2026; 235: 1–12. DOI: <https://doi.org/10.1016/j.jss.2025.112758>.
8. Menke, J., Homberg, S. & Koch, O. “Introduction to artificial intelligence and deep learning using interactive electronic programming notebooks”. *Archiv der Pharmazie*. 2023; 356 (7): 1–10. DOI: <https://doi.org/10.1002/ardp.202200628>.
9. Chattopadhyay, S., Prasad, I., Henley, A.Z., Sarma, A. & Barik, T. “What’s wrong with computational notebooks? Pain Points, Needs, and Design opportunities”. *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020; 1–12. DOI: <https://doi.org/10.1145/3313831.3376729>.
10. Pimentel, J. F., Murta, L., Braganholo, V. & Freire, J. “Understanding and improving the quality and reproducibility of Jupyter notebooks”. *Empirical Software Engineering*. 2021; 26 (65): 1–20. DOI: <https://doi.org/10.1007/s10664-021-09961-9>.
11. Samuel, S. & Mietchen, D. “Computational reproducibility of Jupyter notebooks from biomedical publications”. *GigaScience*. 2024; 13: 1–23. DOI: <https://doi.org/10.1093/gigascience/giad113>.
12. Hossain, A. S. M. S., Brown, C., Koop, D. & Malik, T. “Similarity-Based Assessment of Computational Reproducibility in Jupyter Notebooks”. *ACM REP '25: Proceedings of the 3rd ACM Conference on Reproducibility and Replicability*. 2025. p. 157–167. DOI: <https://doi.org/10.1145/3736731.3746159>.
13. Erickson, C., Wexler, A. & Largent, E. “Digital biomarkers for neurodegenerative disease”. *JAMA Neurology*. 2025; 82 (1): 5–6. DOI: <https://doi.org/10.1001/jamaneurol.2024.3533>.
14. Band, T. G., Bar-Or, R. Z. & Ben-Ami, E. “Advancements in eye movement measurement technologies for assessing neurodegenerative diseases”. *Frontiers in Digital Health*. 2024; 6: 1–6. DOI: <https://doi.org/10.3389/fdgth.2024.1423790>.
15. Graham, L., Vitorio, R., Walker, R., Barry, G., Godfrey, A., Morris, R. & Stuart, S. “Digital Eye-Movement Outcomes (DEMOs) as biomarkers for neurological conditions: A narrative review”. *Big Data and Cognitive Computing*. 2024; 8 (12): 1–21. DOI: <https://doi.org/10.3390/bdcc8120198>.
16. Wolf, A., Tripanpitak, K., Umeda, S. & Otake-Matsuura, M. “Eye-tracking paradigms for the assessment of mild cognitive impairment: a systematic review”. *Frontiers in Psychology*. 2023; 14: 1–24. DOI: <https://doi.org/10.3389/fpsyg.2023.1197567>.
17. Ілуща, А. С. “Веб-орієнтовані програмні засоби інтеграції PaaS і SaaS для підвищення продуктивності дослідницької та освітньої діяльності”. *Вчені записки ТНУ імені В.І. Вернадського. Серія: Технічні науки*. 2026; 37 (76): 97–107. DOI: <https://doi.org/10.32782/2663-5941/2026.2.1/15>.
18. Pavlenko, V., Ilutsa, A. & Kravchenko, Y. “Eye-tracker signals processing in system identification of human oculomotor apparatus with using cloud technologies”. *WSEAS Transactions on Signal Processing*. 2024; 20: 125–137. DOI: <https://doi.org/10.37394/232014.2024.20.13>.
19. Pavlenko, V., Ilutsa, A. & Gidulian, V. “Software tools for organizing cloud computing in psychophysiological research based on eye-tracking data”. *Herald of Advanced Information Technology*. 2024; 7 (4): 394–404. DOI: <https://doi.org/10.15276/hait.07.2024.28>.

20. Pavlenko, V., Lukashuk, D. & Ilutsa, A. "Assessment of psychophysiological status using a cloud computing platform". *SWorldJournal*. 2025; 33 (1): 196–207. DOI: <https://doi.org/10.30888/2663-5712.2025-33-01-124>.

Отримано 06.07.2026

Отримано після перегляду 27.08.2026

Прийнято 04.09.2026

DOI: <https://doi.org/10.15276/ict.03.2026.06>

UDC 004.942

A templated language for transforming cloud computing results for language-independent visualization

Andrii S. Ilutsa

ORCID: <https://orcid.org/0009-0007-3445-8481>; ilutsa.a.s@op.edu.ua. Scopus Author ID: 59735462200

Vitaliy D. Pavlenko

ORCID: <https://orcid.org/0000-0002-5655-4171>; pavlenko_vitalij@ukr.net. Scopus Author ID: 54401442600

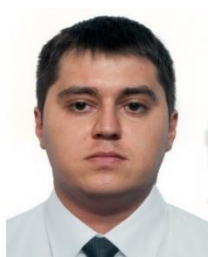
¹⁾ Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

The aim of the work is developing a specialized template language for cloud computing results transformations, which provides their structuring and visualization regardless of the programming language of the tools. The need for such a solution is due to the fact that in traditional approaches, calculation, analysis and graphical representation of results are often implemented in the same program code, therefore, changing the form of presentation requires its editing. It is proposed to separate the computational procedure from the visualization of results. The language is presented as a formalized system of syntactic constructs and rules for their interpretation by the translator. The syntax includes control lines, visualization blocks, data series and service symbols for describing graphical and tabular representations. The translation process includes checking the structure and content of the template, forming an internal hierarchical representation and preparing data for visualization modules. The internal representation is presented as a structure similar to an abstract syntax tree, in which visualization blocks, data series and display parameters have a formalized representation. The developed language provides the transformation of results into standardized structures, unifies their representation and simplifies further analysis in a web-oriented cloud environment. The practical context of the application of the developed language is neurophysiological research based on eye tracking data, in which the results of processing and modeling require further structured representation and visualization.

Keywords: cloud computing, web-oriented software, specialized language, transformation template language, translator, translation, data visualization, eye tracking

ІНФОРМАЦІЯ ПРО АВТОРІВ



Андрій Сергійович - аспірант кафедри Комп'ютеризованих систем та програмних технологій. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Галузь досліджень: WEB-розробка, хмарні обчислення

Andrii S. Ilutsa - PHD student, Department of Computerized Systems and Software Technologies. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ORCID ID: <https://orcid.org/0009-0007-3445-8481>; ilutsa.a.s@op.edu.ua. Scopus Author ID: 59735462200

Research field: WEB development, cloud computing



Павленко Віталій Данилович – доктор технічних наук, професор кафедри Комп'ютеризованих систем та програмних технологій. Національний університет «Одеська політехніка», пр. Шевченка, 1.. Одеса, 65044, Україна

Галузь досліджень: інтелектуальні інформаційні технології, непараметрична ідентифікація, моделі Вольтерри

Vitaliy D. Pavlenko - Doctor of Engineering Science, Professor, Department of Computerized Systems and Software Technologies. . Odesa Polytechnic National University, 1.,Shevchenko Ave. Odesa, 65044, Ukraine

ORCID ID: <https://orcid.org/0000-0002-5655-4171>; pavlenko_vitalij@ukr.net. Scopus Author ID: 54401442600

Research field: intelligent information technologies, nonparametric identification, Volterra models