УДК 004.4'2 https://doi.org/10.15407/pp2019.01.037

О.М. Овдій

ДО ПИТАННЯ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ РОБОЧИХ ПРОЦЕСІВ НА ОСНОВІ АЛГЕБРО-АЛГОРИТМІЧНОГО ТА ОНТОЛОГІЧНОГО ІНСТРУМЕНТАРІЮ

Представлено концепцію системи для автоматизації проектування робочих процесів на основі алгебро-алгоритмічного та онтологічного інструментарію. Проведено аналіз існуючих підходів та запропоновано архітектуру системи. Роботу системи проілюстровано на прикладі розробки робочого процесу для аналізу великих обсягів даних на розподіленій платформі Apache Hadoop. Показано, що поєднання інструментів онтологій та алгебро-алгоритмічних інструментів забезпечує значний потенціал для адаптації, оптимізації, інтеграції та модифікації. Результати кількісної оцінки ефективності запропонованих засобів свідчать про те, що вони здатні сприяти істотному підвищенню продуктивності розробки і зменшенню трудовитрат.

Ключові слова: робочі процеси, онтологія, алгебра алгоритмів, проектування і синтез програм, розподілені обчислення.

Вступ

В останні роки активно розвиваються та розповсюджуються розподілені обчислювальні платформи. Зокрема це пов’язано з величезною кількістю цифрових даних, які потребують обробки і обсяг цих даних стрімко зростає з кожним днем. Поряд з цим, із зростом можливостей зростає і складність систем, а відповідно складність розробки програм для них. Сучасні програмні системи зазвичай є комплексними та складаються з різнорідних сервісів, що реалізовані на базі різних технологій, різними мовами програмування та розподілені у мережі. Крім того програми функціонують у гетерогенному середовищі, що швидко еволюціонує, інфраструктура змінюється, сервіси з’являються, змінюються та зникають. Одним з інструментів для спрощення проектування та підтримки таких програмних систем є робочі процеси (workflow) та системи керування ними. Формальна концепція робочого процесу існує вже протягом тривалого часу. Згідно з визначенням Workflow Management Coalition (WfMC) [1] робочий процес це автоматизація бізнес-процесу, цілком або частково, протягом якого документи, інформація або завдання передаються від одного учасника до іншого для дії, згідно з набором процедурних правил. У відношенні до розподілених обчислювальних платформ робочі процеси можна визначити як оркестрацію набору дій для досягнення більшої і складнішої мети [2]. Сьогодні системи керування робочими процесами широко використовуються як у бізнесі, так і у науковому секторі. Робочі процеси застосовуються в багатьох наукових дисциплінах, часто використовуючи великі за обсягом та різноманітні ресурси даних, а також паралельні та розподілені обчислювальні платформи. Робочі процеси забезпечують систематичний спосіб опису необхідних методів та інтерфейс між фахівцями та обчислювальним середовищем. Завдяки різкому збільшенню обсягів первинних даних робочі процеси відіграють все більш важливу роль, дозволяючи науковцям формулювати методи обробки та аналізу даних з різних джерел на широкому спектрі обчислювальних платформ. Наукова діяльності є дослідницькою за своїм характером і часто виконується методом "проб і помилок" а це потребує ще більшої адаптації. Крім того предметні експерти та науковці зазвичай не є професійними програмістами і їм потрібні інструменти, якими вони зможуть з легкістю користуватися. В зв’язку з цим виникає необхідність створення спеціальних високорівневих засобів проектування робочих процесів.

У даній роботі запропоновано концепцію системи для автоматизації проектування робочих процесів на основі алгебро-алгоритмічного та онтологічного інструментарію ОКРП. Дана система є подальшим розвитком методологій та інструментів, що ґрунтуються на засобах високорівневої алгебро-алгоритмічної формалізації і автоматизації перетворень програм [2–10]. Зокрема експериментальної інструментальної системи ІПС для конструювання та оптимізації паралельних програм, а також її онлайн версії системи ОДСП.

Основна відмінність інструментарію ОКРП у порівнянні з попередніми системами полягає в орієнтації на проектування робочих процесів та зберігання розроблюваних робочих процесів як екземплярів онтології у форматі OWL [11].

1. Підхід до проектування робочих процесів на основі онтологій і засобів алгебри алгоритмів

Одним з перспективних напрямів у розробці та дослідженні систем паралельних і розподілених обчислень нині є побудова програмних абстракцій у вигляді алгебро-алгоритмічних мов і моделей. У роботах [2–10] запропоновані теорія, методологія та інструментарій для автоматизованого проектування паралельних програм, що ґрунтуються на засобах високорівневої алгебро-алгоритмічної формалізації та автоматизації перетворень програм.

Наразі розроблено кілька алгоритмічних алгебр, найвідомішими з яких є алгебра Дейкстри, алгебра схем Янова та алгоритміка програм, досліджена у працях В.М. Глушкова та його учнів. Використання систем алгоритмічних алгебр (САА) В.М. Глушкова надає потужні можливості для розвитку архітектурно- і мовно-незалежних засобів програмування для розподілених обчислювальних систем і мереж.

Представлення алгоритмів алгебро-алгоритмічними засобами дозволяє автоматизовано їх перетворювати та оптимізувати, а крім того представляти алгоритми у природно-лінгвістичній формі, що є більш зрозумілою.

Онтологічні засоби, в свою чергу, представляють знання семантично, забезпечують необхідний рівень абстракції, сприяють їх поширенню та полегшують інтеграцію з іншими системами.

Онтологія [12], яка спочатку була введена в IT як засіб забезпечення семантики в Semantic Web, нині широко застосовується для забезпечення семантики та механізмів комунікації, та структурування знань у різноманітних галузях IT, бізнесу та багатьох інших сферах людської діяльності. Онтології нині застосовуються фактично на всіх етапах життєвого циклу програмного забезпечення [13]. Зокрема онтології застосовуються для вимог інженерії, представлення об’єктної моделі домену, документування коду, опису бізнес-правил, тестування та ін.

Онтології можуть:

– додати необхідний рівень абстракції, що спростить перетворення та адаптацію алгоритмів;

– посприяти структуруванню знань;

– забезпечити механізм комунікації та інтеграції.

Аналіз існуючих онтологічних підходів до інженерії програмного забезпечення, а зокрема робочих процесів виявив відсутність загальноприйнятих стандартів. В рамках роботи проаналізовано такі онтології як OWL-WS [14], SciFlow [15], WDO-It [16] та ін. В ході аналізу виявлено недоліки існуючих підходів:

– недостатній рівень абстракції;

– вузька спеціалізація на конкретному домені;

– спроба забезпечити високий рівень абстракції, поряд з наслідуванням спеціалізованих онтологій, що були розроблені для інших цілей.

У зв'язку з переліченими вище недоліками у даній роботі замість використання існуючих онтологій було вирішено розроблену раніше прикладну онтологію для проектування програм [10] розширити для проектування робочих процесів. З метою подальшого розвитку інструментарію та для збільшення переліку підтримуваних платформ онтологія побудована з належним рівнем абстракції. Як інструменталь-

ний засіб розробки онтології обрана система Protégé [17], та використано мову опису онтологій OWL (Web Ontology Language) [11].

На рис. 1 показано фрагмент ієрархії концептів розробленої онтології. Вузлами графу є концепти, а дугами відношення наслідування між ними.

Підкласи класу Data представляють різні структури даних. Клас Construction відображає типи конструкцій: атомарні (AtomicConstruction) та складні (CompoundConstruction). Поняття складної конструкції відповідає поняттю вкладеного робочого процесу. Клас Operation відображає типи операції САА: оператори та предикати. Операції можуть бути базисними (BasicOperation) або складними (CompoundOperation). Класи Construction

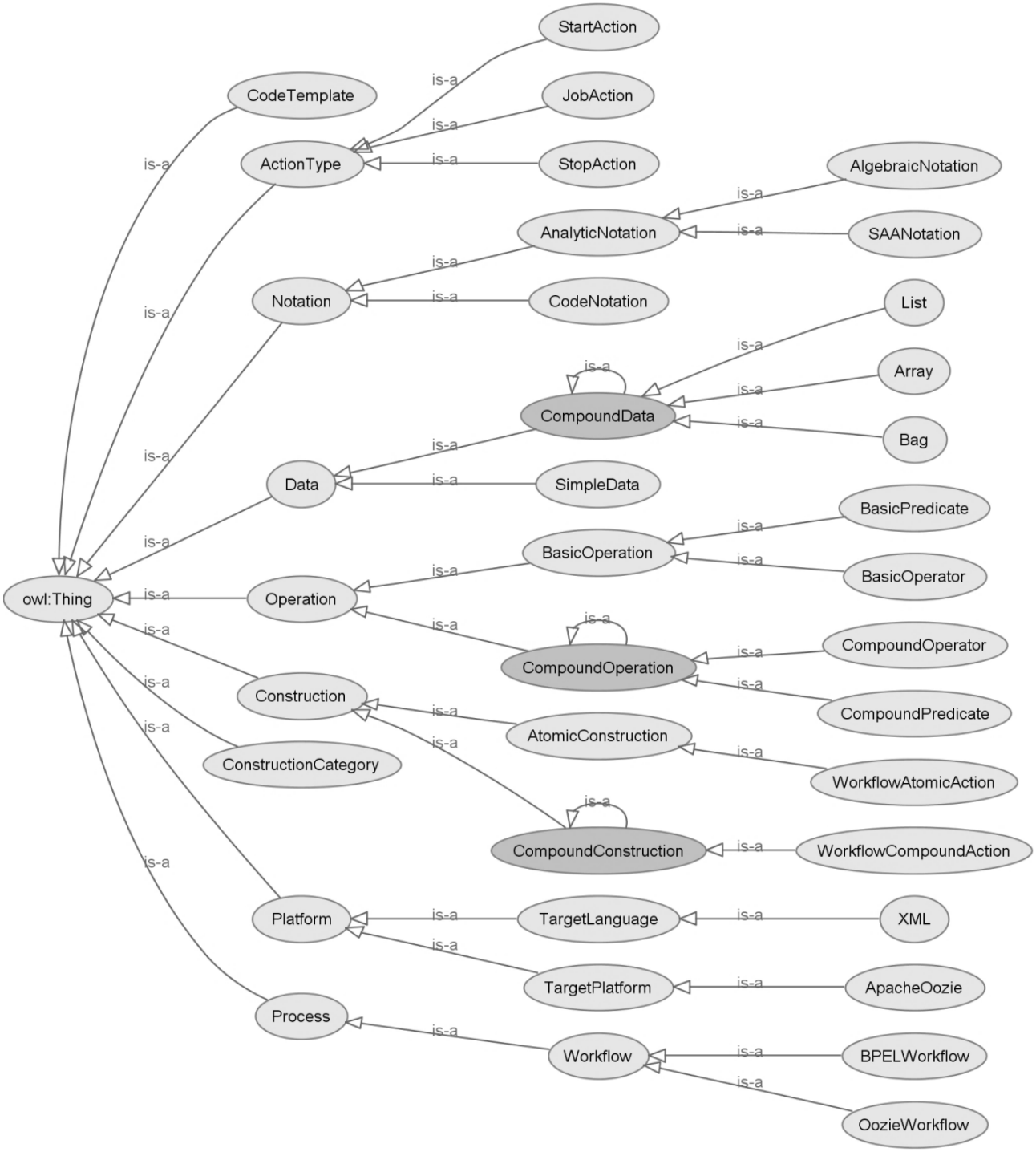

Рис. 1. Фрагмент ієрархії концептів онтології проектування робочих процесів

мають властивість isOperation, що ставить їм відповідний клас Operation для того щоб зв'язати конструкції робочого процесу з відповідними типами операцій в CAA. Клас Platform представляє підтримувані платформи та мови. ActionType – визначає тип дії робочого процесу, а саме може вона мати і вхідні і вихідні зв'язки, або тільки вхідні чи вихідні. Клас Notation представляє типи нотацій: алгебраїчну (AlgebraicNotation), природньо-лінгвістичну (SAANotation) та мовою програмування (CodeNotation). Клас Process призначений для специфікації алгоритмів, зокрема робочих процесів.

Включення семантики до специфікації робочого процесу дозволяє відокремлювати абстрактну специфікацію від фактичної реалізації, роблячи можливим переніс робочого процесу з однієї платформи керування робочими процесами на іншу та спрощує адаптацію до змін у середовищі виконання. Крім того, у зв'язку з специфікою наукових досліджень, наукові робочі процеси часто модифікуються та повторно використовуються.

З рухом у напрямку відкритої науки, виникає потреба зробити робочі процеси доступними не тільки їх авторам, а й іншим зацікавленим сторонам з різним рівнем підготовки від аматорів до професіоналів. Використання інструментів онтологій робити робочий процес більш зрозумілим для інших людей та програмних агентів.

В свою чергу, використання інструментів алгебри алгоритмів теж сприяє підвищенню рівня абстракції та підтримці гнучкості та платформо-незалежності. Окрім того дозволяє автоматизувати перетворення та оптимізацію робочих процесів.

Таким чином поєднання інструментів онтологій та алгебро-алгоритмічних інструментів забезпечує значний потенціал для адаптації, оптимізації, інтеграції та модифікації розроблюваних робочих процесів.

2. Онлайновий конструктор робочих процесів

В даній роботі представлено концепцію системи для автоматизації проектування робочих процесів на основі алгебро-алгоритмічного та онтологічного інструментарію: онлайновий конструктор робочих процесів (ОКРП).

Розробка даного інструменту була викликана необхідністю у подальшому розвитку інструментів автоматизованого проектування програм для підтримки проектування алгоритмів, що кодуються не звичайними мовами програмування, а визначаюся, наприклад, XML-специфікацією або спеціалізованими мовами програмування, зокрема робочих процесів.

В розроблюваній системі ОКРП, алгоритми робочих процесів представлені у графічній, онтологічній, аналітичній та природньо-лінгвістичній формах. Представлення алгоритму у аналітичній формі дозволяє виконувати перетворення, що може знадобитися для пристосування до змін у платформі або оптимізації. Графічна та природньо-лінгвістична форми роблять процес проектування алгоритмів зручним для людини, що полегшує досягнення необхідної якості програм, підвищує продуктивність та дозволяє уникати синтаксичних помилок. Онтологічне представлення та зберігання алгоритмів семантично їх збагачує, сприяє структуруванню знань та забезпечує механізм комунікації та інтеграції.

Основні особливості ОКРП:

– орієнтація на конструювання алгоритмів для робочих процесів та спеціалізованих мов програмування;

– графічний конструктор алгоритмів робочих процесів;

– форми представлення алгоритмів:

  - графічна,
  - аналітична,
  - природно-лінгвістична;

– зберігання розроблюваних алгоритмів як екземплярів онтології у форматі OWL.

Автоматизована система ОКРП є онлайновим сервісом, що призначений для проектування та генерації описів робочих процесів на основі високорівневих специфікацій алгоритмів. Система спрямована забезпечувати побудову робочих процесів у режимі порівневого конструювання та генерацію їх специфікації.

ОКРП ґрунтується на використанні розглянутих засобів CAA-M та онтологій.

Графічний інтерфейс діалогового конструктора орієнтований на зручність та легкість конструювання та виключення можливості появи синтаксичних помилок у процесі побудови алгоритму. Система базується на сервісно-орієнтованій архітектурі та спрямована на багатокористувальницьке використання інструментарію через мережу Інтернет. Розробка ведеться мовою Java.

Розроблювана автоматизована система ОКРП складається з компонентів, показаних на рис. 2.

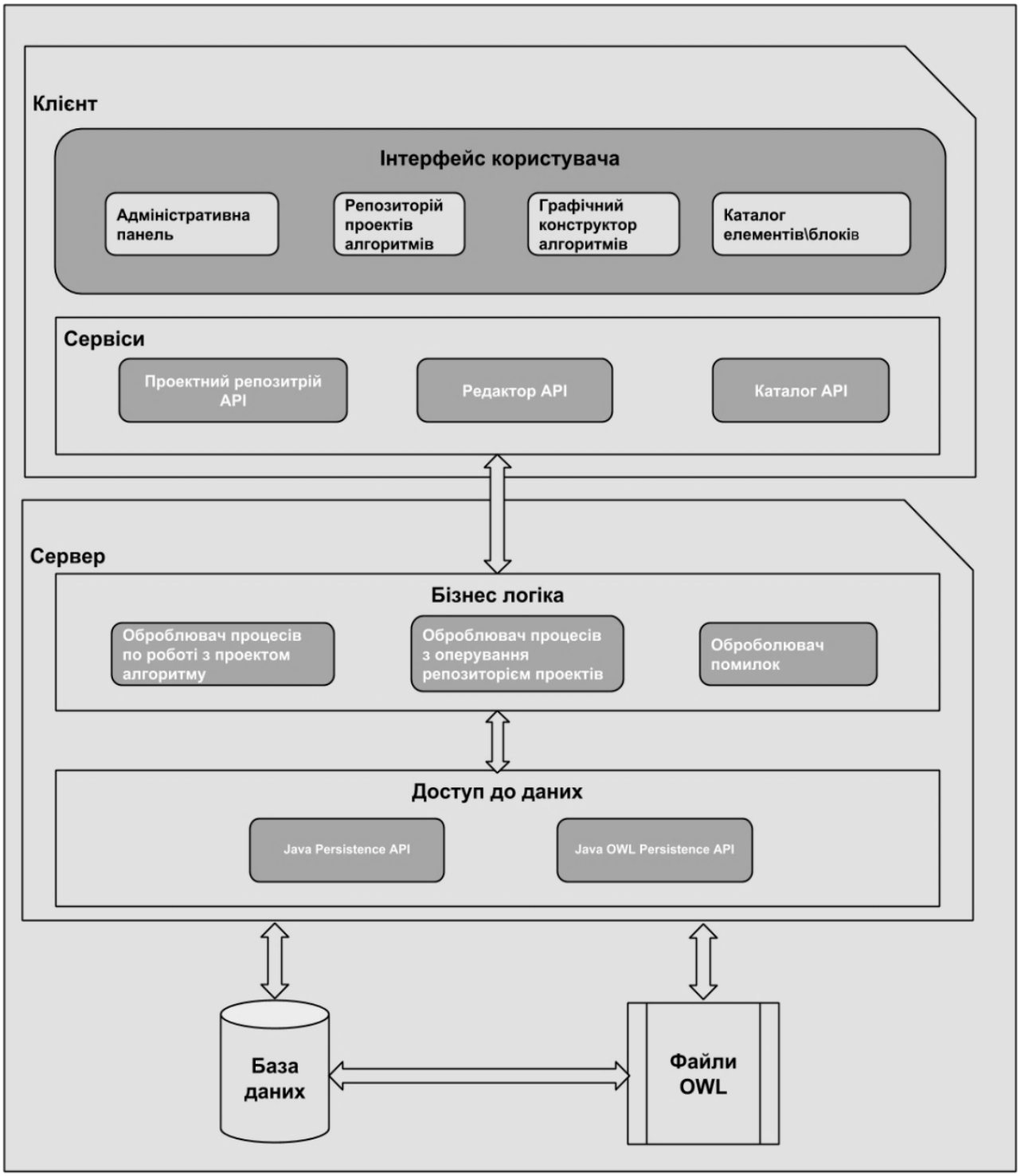

Рис. 2. Архітектура системи проектування робочих процесів

3. Приклад використання системи ОКРП

Задача обробки та аналізу великих обсягів даних надзвичайно актуальна у сфері метеорології. В ній наявні величезні обсяги історичних та поточних даних, а моделі та засоби аналізу надзвичайно складні та ресурсоємні. Наприклад, Національне управління океанічних і атмосферних досліджень США (NOAA) [18] щодня видає близько 20 терабайт даних, включаючи дані сотень метеорологічних станцій, результати прогнозів, дані супутників тощо. Для роботи з такими великими наборами даних науковцям необхідні інструменти, якими вони можуть користуватися без поглиблених знань програмування та архітектури обчислювальних систем.

Розглянемо функціонування системи ОКРП на прикладі розробки робочого процесу для аналізу великих обсягів даних на розподіленій платформі Apache Hadoop [19]. Apache Hadoop призначена для розподіленого зберігання й обробки великих обсягів даних на великих комп'ютерних кластерах, реалізація парадигми MapReduce. Робочий процес проектується для системи Apache Oozie [20]. Apache Oozie є системою керування робочими процесами для керування роботами Apache Hadoop. Для попередньої обробки даних використовується Apache Pig [21], що призначений для обробки великих наборів даних на Apache Hadoop. Для аналізу даних застосовується програмне середовище R [22] для статистичних обчислень та аналізу даних.

На рис. 3 показано інтерфейс розроблюваної системи з прикладом спроектованого алгоритму для робочого процесу Apache Oozie статистичного аналізу часових рядів у метеорологічних даних.

Робочий процес складається з наступних кроків:

– **Start** – запуск робочого процесу;

– **Load ncdc data** – вузол робочого процесу, який виконує запуск shell-скрипту, що завантажує сирі історичні данні метеорологічних станцій з сайту національного центру кліматичних даних США NCDC [18];

– **Prepare data** – запускає програму Apache Pig, що виконує попередню обробку даних. Виконується фільтрація даних за розташуванням метеостанції (м. Київ) та обчислюється середньомісячна температура за останні декілька років. Детальний приклад обробки даного масиву метеорологічних даних за допомогою Apache Pig розглянуто у попередній роботі [23].

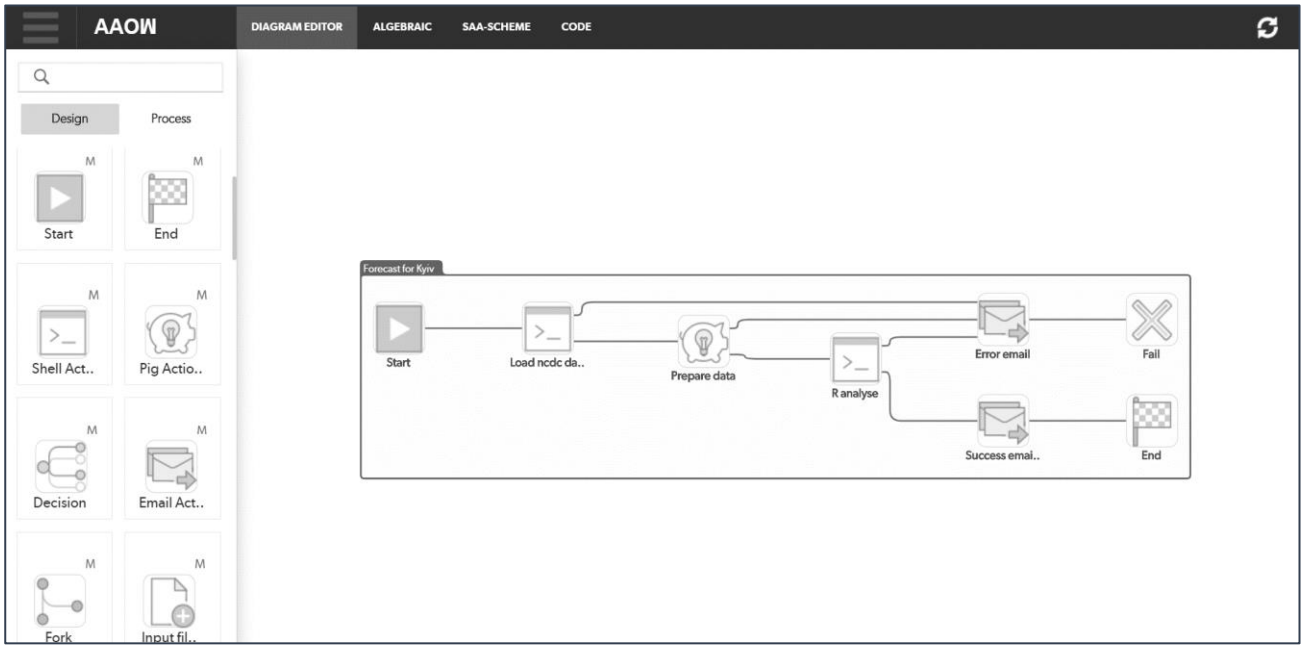

Рис. 3. Фрагмент копії екрану системи ОКРП

– **R analyse** – shell-скрипт який запускає на виконання програму R для статистичного аналізу даних та виводу результатів. Програма обчислює прогноз середньомісячної температури в м. Києві на наступні два роки за моделлю аналізу часових рядів ARIMA та графічно представляє результат.

– **Error email** – сповіщення у разі виникнення помилок та збою робочого процесу.

– **Success email** – сповіщення про вдале виконання робочого процесу та надсилання результатів.

Як кроки робочого процесу можна використовувати вже існуючі програми та сервіси або, при необхідності, розробити нові, зокрема за допомогою систем ІПС або ОДСП. Наприклад, програма Apache Pig, що використовується у кроці **Prepare data**, була розроблена у попередній роботі за допомогою системи ОДСП [23].

Далі наведено скорочений приклад алгебраїчного представлення спроектованого алгоритму для робочого процесу статистичного аналізу часових рядів у метеорологічних даних:

MID_TEMP_ANALYSE = Start THEN
Shell_action(upload-data, load_ncdc.sh, arguments, file) THEN
Pig_action(pig-prepare-data, prepare_data.pig, params, file) THEN
Shell_action(shell-r-analyse, run_r_hadoop.sh, arguments, file)
THEN End(end)

де **Start** – початок робочого процесу; **Shell_action(upload-data..** – вузол робочого процесу, який виконує запуск shell-скрипту, що завантажує сирі дані метеорологічних станцій; **Pig_action(pig-prepare-data..** – запускає програму Apache Pig, що виконує попередню фільтрацію та обробку даних; **Shell_action(shell-r-analyse..** - shell-скрипт, який запускає на виконання програму R для статистичного аналізу даних та виводу результатів.

Далі наведено приклад природно-лінгвістичного представлення попереднього алгоритму:

"MID_TEMP_ANALYSE"
==== "Start workflow"
THEN
"Run Shell Action upload-data with parameters: (load_ncdc.sh, arguments, file)"
THEN
"Run Pig Action pig-prepare-data with parameters (prepare_data.pig, params, file)"
THEN
"Run Shell Action shell-r-analyse with parameters: (run_r_hadoop.sh, arguments, file)"
THEN
"End of workflow (end)"

Далі системою буде згенеровано код специфікації робочого процесу для цільової системи керування робочими процесами, у даному випадку Apache Oozie (рис. 4).

Припустимо, що у постачальника метеорологічних даних змінився формат представлення даних. Для вирішення цієї проблеми в існуючий робочий процес після кроку **Load ncdc data** необхідно додати ще один крок, який буде виконувати конвертацію даних з нового формату. Інші кроки та їх взаємозв'язки не зміняться і для розробки та додавання нового кроку не потрібно вникати в деталі реалізації та функціонування всього робочого процесу, достатньо буде знати вимоги до формату даних. Крім того, використовуючи інструменти алгебри алгоритмів, можна буде автоматизовано перетворити всі існуючі робочі процеси та автоматично додати необхідний крок конвертації даних.

З наведеного прикладу видно, що робочі процеси для розподілених та гетерогенних середовищ з легкістю можна

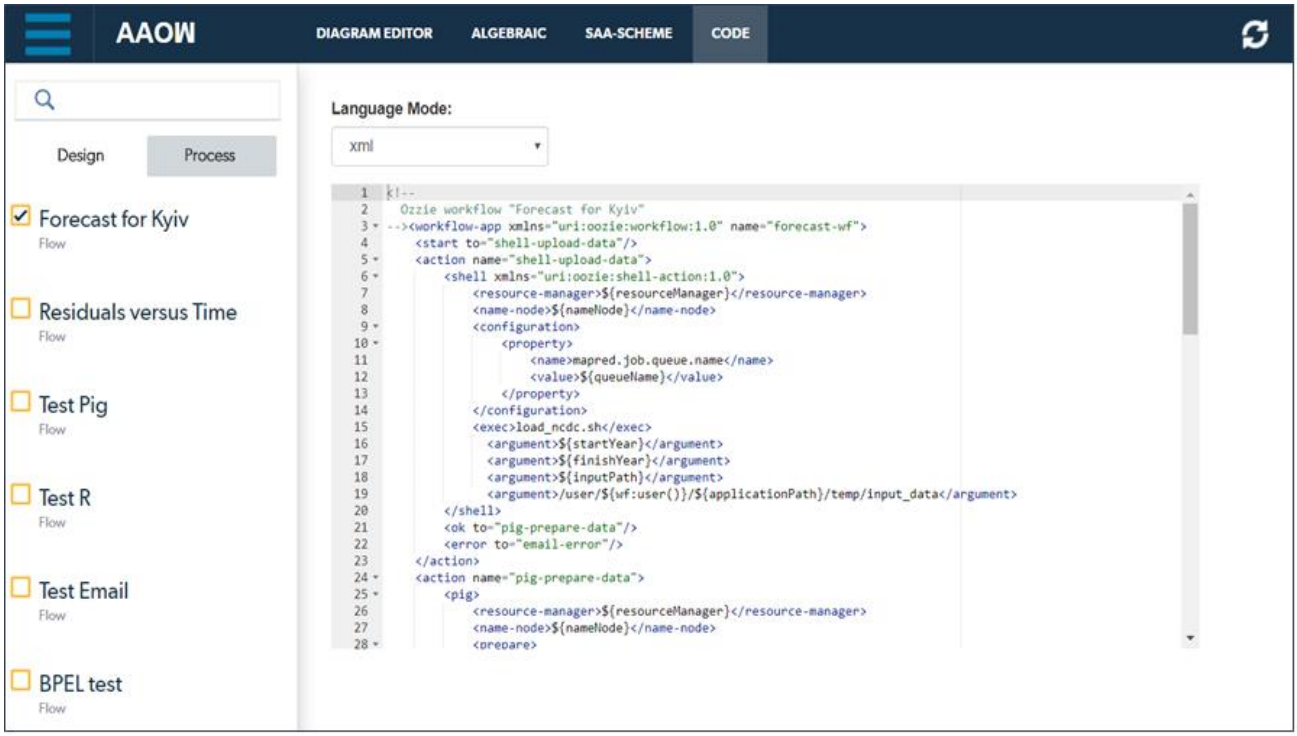

Рис. 4. Копія екрану системи ОКРП зі згенерованим кодом специфікації робочого процесу

проектувати та модифікувати за допомогою інструменту проектування робочих процесів ОКРП.

Зменшення обсягу коду необхідного для реалізації задачі зменшує трудовитрати на кодування і подальшу підтримку, та скорочує число можливих помилок. Длякількісної оцінки ефективності запропонованих засобів була використана метрика SLOC (Source Lines of Code, кількість рядків коду) [24], широко використовувана для оцінки трудовитрат на розробку програм. Для наведеного прикладу результати оцінки SLOC для САА-схеми і згенерованого за нею xml коду специфікації робочого процесу для цільової платформи Apache Oozie складає 16 і 88 відповідно. Таким чином, співвідношення рядків коду показує, що обсяг цільового коду перевищує обсяг САА-схеми більш ніж в 5 разів. Дані результати свідчать про те, що запропоновані засоби для автоматизованого проектування робочих процесів здатні сприяти підвищенню продуктивності розробки і зменшенню трудовитрат.

Висновки

В даній роботі представлено концепцію системи для автоматизації проектування робочих процесів на основі алгебро-алгоритмічного та онтологічного інструментарію: онлайновий конструктор робочих процесів ОКРП. Проведено аналіз існуючих підходів та запропоновано архітектуру системи. Роботу системи проілюстровано на прикладі розробки робочого процесу Apache Oozie для аналізу великих обсягів даних на розподіленій платформі Apache Hadoop. Результати кількісної оцінки ефективності запропонованих засобів свідчать про те, що вони здатні сприяти істотному підвищенню продуктивності розробки і зменшенню трудовитрат. Далі планується подальша розробка та розвиток системи ОКРП, зокрема розширення підтримкою інших стандартів та систем керування робочими процесами, зокрема BPEL.

Література

1. Workflow Management Coalition: сайт. URL: https://www.wfmc.org/ (дата звернення: 29.01.2019).

2. Amin K., Laszewski G., Hategan M., Zaluzec N. J., Hampton S., Rossi A. GridAnt: A Client-Controllable Grid Workflow System. In Proc. of the 37th Hawaii International Conference on System Sciences, HI, USA, 2004. P. 3293–3301.
3. Дорошенко А.Ю., Бекетов О.Г., Іванів Р.Б., Іовчев В.О., Мироненко І.О., Яценко О.А. Автоматизована генерація паралельних програм для графічних прискорювачів на основі схем алгоритмів. *Проблеми програмування.* 2015. № 1. С. 19–28.
4. Андон Ф.И., Дорошенко А.Е., Бекетов А.Г., Иовчев В.А., Яценко Е.А. Инструментальные средства автоматизации параллельного программирования на основе алгебры алгоритмов. *Кибернетика и системный анализ.* 2015. № 1. С. 162–170.
5. Дорошенко А.Ю., Іваненко П.А., Овдій О.М., Яценко О.А. Автоматизоване проектування програм для розв'язання задачі метеорологічного прогнозування. *Проблеми програмування.* 2016. № 1. С. 102–115.
6. Андон Ф.И., Дорошенко А.Е., Цейтлин Г.Е., Яценко Е.А. Алгеброалгоритмические модели и методы параллельного программирования. Киев: Академпериодика, 2007. 631 с.
7. Дорошенко Е.А., Яценко Е.А. О синтезе программ на языке Java по алгеброалгоритмическим спецификациям. *Проблеми програмування.* 2006. № 4. С. 58–70.
8. Яценко Е.А. Интеграция средств алгебры алгоритмов и переписывания термов для разработки эффективных параллельных программ. *Проблеми програмування*. 2013. № 2. С. 62–70.
9. Дорошенко А.Ю., Бекетов О.Г., Яценко О.А, Вітряк Є.А., Павлючин Т.О. Розробка сервісно-орієнтованих засобів для запуску паралельних програм на мультипроцесорному кластері. *Проблеми програмування*. 2014. № 4. С. 3–14.
10. Дорошенко Е.А., Овдей О.М., Яценко Е.А. Онтологические и алгеброалгоритмические средства автоматизации проектирования параллельных программ для "облачных" платформ. *Кибернетика и системный анализ.* 2017. Т. 53, № 2. С. 181–192.
11. OWL 2 Web Ontology Language Primer (Second Edition). URL: https://www.w3.org/2012/pdf/REC-owl2-primer-20121211.pdf (дата звернення: 29.01.2019).
12. Gruber T. R. A Translation Approach to Portable Ontologies. Knowledge Acquisition. 1993. N 5(2). P. 199–220.
13. Strmecki D., Magdalenic I., Kermek D. An Overview on the use of Ontologies in Software Engineering. *Journal of Computer Science* 2016. N 12(12). P. 597–610.
14. Beco S., Cantalupo B., Giammarino L., Matskanis N., Surridge M. OWL-WS: A Workflow Ontology for Dynamic Grid Service Composition. In: 1st Int. Conf. on e-Science and Grid Computing. IEEE Computer Society. 2005. P. 148–155.
15. Oliveira D., Ogasawara E., Araujo Baiao F., Mattoso M. Adding Ontologies to Scientific Workflow Composition. XXVI Simpósio Brasileiro de Banco de Dados. 2011. Florianópolis, SC. P. 147–154.
16. Pinheiro da Silva P., Salayandia L., Gates A. Q. WDO-It! A Tool for Building Scientific Workflows from Ontologies. Technical Report UTEP-CS-07-XX, University of Texas. 2007. URL: http://digitalcommons.utep.edu/ cs_techrep/201 (дата звернення: 29.01.2019).
17. Horridge M. A practical guide to building OWL ontologies using Protégé 4 and CO-ODE tools. Manchester: The University Of Manchester. 2011. 107 p.
18. National Climatic Data Center (NCDC): сайт. URL: https://www.ncdc.noaa.gov/ (дата звернення: 26.02.2018).
19. Apache Hadoop: сайт. URL: http://hadoop.apache.org/ (дата звернення: 29.01.2019).
20. Apache Oozie Workflow Scheduler for Hadoop: сайт. URL: http://oozie.apache.org/ (дата звернення: 29.01.2019).
21. Apache Pig: сайт. URL: http://pig.apache.org/ (дата звернення: 29.01.2019).
22. The R Project for Statistical Computing: сайт. URL: https://www.r-project.org/ (дата звернення: 29.01.2019).
23. Овдій О.М. Розширення системи синтезу програм з метою аналізу великих наборів даних. *Проблеми програмування* 2018. № 2-3. Спец. Вип. С. 68–74.
24. Nguyen V., Deeds-Rubin S., Tan T., Boehm B. A. SLOC Counting Standard. URL: http://csse.usc.edu/TECHRPTS/2007/usc-csse-2007-737/usc-csse-2007-737.pdf. (дата звернення: 29.01.2019).

References

1. Wfmc.org. Workflow Management Coalition. [online] Available from: https://www.wfmc.org/ [Accessed 29 Jan. 2019].
2. Amin, K., Laszewski, G., Hategan, M., Zaluzec, N. J., Hampton, S. & Rossi, A. (2004). GridAnt: A Client-Controllable Grid Workflow System. In: *37th Hawaii International Conference on System Sciences*. HI, USA. P. 3293–3301.
3. Doroshenko A.Yu., Beketov O.G., Ivaniv R.B., Iovchev V.O., Myronenko I.O. & Yatsenko O.A. (2015) Automated generation of parallel programs for graphics processing units based on algorithm schemes. *Problems in programming*. (1). P. 19–28. (in Ukrainian).
4. Andon P.I., Doroshenko A.Yu., Beketov O.G., Iovchev V.O. & Yatsenko O.A. (2015) Software tools for automation of parallel programming on the basis of algebra of algorithms. *Cybernetics and systems analysis*. (1). P. 162–170. (in Russian).
5. Doroshenko A.Yu., Ivanenko P.A., Ovdii O.M., & Yatsenko O.A. (2016) Automated design of programs for solving the task of meteorological forecasting. *Problems in programming*. (1). P. 102–115. (in Ukrainian).
6. Andon P.I. et al. (2007) *Algebra-algorithmic models and methods of parallel programming*. Kiev: Academperiodika. (in Russian).
7. Doroshenko A.Yu. & Yatsenko O.A. (2006) About the synthesis of Java programs by algebra-algorithmic specifications. *Problems in programming*. (4). P. 58–70. (in Russian).
8. Yatsenko O.A. (2013) Integration of algebra-algorithmic tools and term rewriting for efficient parallel programs development. *Problems in programming*. (2). P. 62–70. (in Russian).
9. Doroshenko A.Yu., Beketov O.G. Yatsenko O.A., Pavliuchyn T.O. & Vitriak I.A. (2014) Development of the service-oriented software for launching parallel programs on a multiprocessor cluster. *Problems in programming.* (4). P. 3–14. (in Ukrainian).
10. Doroshenko A.Yu., Ovdii O.M. & Yatsenko O.A. (2017) Ontological and algebra-algorithmic tools for automated design of parallel programs for cloud platforms. *Cybernetics and Systems Analysis*. 53(2). P. 181–192. (in Russian).
11. OWL 2 Web Ontology Language Primer (Second Edition). [online] Available from: https://www.w3.org/2012/pdf/REC-owl2-primer-20121211.pdf [Accessed 29 Jan. 2019].
12. Gruber T.R. (1993) A Translation Approach to Portable Ontologies. *Knowledge Acquisition*. 5(2). P. 199–220.
13. Strmecki D., Magdalenic I. & Kermek D. (2016) An Overview on the use of Ontologies in Software Engineering. *Journal of Computer Science*. 12(12). P. 597–610.
14. Beco S., Cantalupo B., Giammarino L., Matskanis N. & Surridge M. (2005) OWL-WS: A Workflow Ontology for Dynamic Grid Service Composition. In: *1st Int. Conf. on e-Science and Grid Computing*. IEEE Computer Society. P. 148–155.
15. Oliveira D., Ogasawara E., Araujo Baiao F. & Mattoso M. (2011) Adding Ontologies to Scientific Workflow Composition. In: *XXVI Simpósio Brasileiro de Banco de Dados.* Florianópolis, SC. P. 147–154.
16. Pinheiro da Silva, P., Salayandia L. & Gates A.Q. (2007) WDO-It! A Tool for Building Scientific Workflows from Ontologies. Technical Report UTEP-CS-07-XX, University of Texas. [online] Available from: http://digitalcommons.utep.edu/cs_techrep/201 [Accessed 29 Jan. 2019].
17. Horridge M. (2011) *A practical guide to building OWL ontologies using Protégé 4 and CO-ODE tools*. Manchester: The University Of Manchester. 2011.
18. Ncdc.gov. *National Climatic Data Center (*NCDC*).* [online] Available from: https://www.ncdc.noaa.gov/ [Accessed 29 Jan. 2019].
19. Hadoop.apache.org. *Apache Hadoop Official Website*. [online] Available from: http://hadoop.apache.org/ [Accessed 29 Jan. 2019].
20. Oozie.apache.org. *Apache Oozie Workflow Scheduler for Hadoop Official Website*. [online] Available from: http://oozie.apache.org/ [Accessed 29 Jan. 2019].
21. Pig.apache.org. *Apache Pig Official Website*. [online] Available from: http://pig.apache.org/ [Accessed 29 Jan. 2019].
22. R-project.org. *The R Project for Statistical Computing Official Website.* [online] Available from: https://www.r-project.org/ [Accessed 29 Jan. 2019].
23. Ovdii, O.M. (2018) Extension of the program synthesis system to analyze large data sets.

Problems in programming. (2-3). P. 68–74. (in Ukrainian).
24. Nguyen V., Deeds-Rubin S., Tan T., Boehm B.A. SLOC Counting Standard. [online] Available from: http://csse.usc.edu/TECHRPTS/2007/usc-csse-2007-737/usc-csse-2007-737.pdf. [Accessed 29 Jan. 2019].

Одержано 31.01.2019

***Про автора*:**

Овдій Ольга Михайлівна,
молодший науковий співробітник.
Кількість наукових публікацій в українських виданнях – 22.
Кількість наукових публікацій в зарубіжних виданнях – 5.
http://orcid.org/0000-0002-8891-7002.

Місце роботи автора:

Інститут програмних систем
НАН України,
03187, м. Київ-187,
проспект Академіка Глушкова, 40.
Тел.: (044) 526 6033.
E-mail: olga.ovdiy@gmail.com