

Г.А. ПІДНЕБЕСНА, молодш. наук. співробітник,
Міжнародний науково-навчальний центр
інформаційних технологій та систем НАН та МОН України,
03187, м. Київ, просп. Академіка Глушкова, 40,
pidnebesna@ukr.net

КОНСТРУЮВАННЯ КОМПЛЕКСУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ІНДУКТИВНОГО МОДЕЛЮВАННЯ З ВИКОРИСТАННЯМ ОНТОЛОГІЙ

Розглянуто процес конструювання комплексу інструментальних засобів МГУА. Особливістю такого комплексу є використання бази знань у вигляді онтології предметної області індуктивного моделювання на основі МГУА. Застосування онтологічного підходу до проектування бази знань при конструюванні програмних засобів побудови моделей за експериментальними даними дає можливість автоматизованого конструювання правил вибору ефективних засобів розв'язання конкретних прикладних задач моделювання складних об'єктів. Це може підвищити рівень інтелектуальності процесу конструювання та спростити і зменшити вартість розроблення та модифікації таких засобів. Фрагменти онтології індуктивного моделювання на основі МГУА наведено як приклад створення формального опису предметної галузі для побудови онтологій використовувався онторедатор Protege_4.3.

Ключові слова: індуктивне моделювання, МГУА, онтологічний підхід, інтелектуальні системи, база знань.

Вступ

З розвитком інтернету, з появою великої кількості інформаційних ресурсів потреба автоматичного оброблення даних призвела до розроблення інтелектуальних засобів аналізу інформації з урахуванням семантики цієї інформації. Основу розробки відповідних інтелектуальних комп'ютерних систем моделювання і прогнозування складних процесів складають результати структурування знань з метою конструювання баз знань на основі онтологій [1], функціонального забезпечення та засобів інтелектуального інтерфейсу.

В умовах сучасного світу при проектуванні інтелектуальних (складних) комп'ютерних систем дедалі більша увага приділяється

моделюванню самих процесів проектування програмного забезпечення. Набуває поширення концепція *MDD (Model Driven Development* – розробка, керована моделями) [2], тобто розробки програм, суть якої полягає в побудові абстрактної метамоделі управління та обміну метаданими і формуванні способів її трансформації. Загальним трендом в дослідженнях автоматизації проектування систем ПЗ є використання методів і засобів онтологічного моделювання як самих процесів проектування, так і специфікацій розроблюваних систем та предметних галузей. Ці дослідження призвели до появи нового сучасного напрямку – *Ontology-Based (-Driven) Software Engineering* – проектування програмного забезпечення на основі онтологій [3]. Онтологічне моделювання дозволяє

змістити акцент з процесу збірки програмно-го забезпечення прикладних систем з великих функціональних блоків до моделей генерації багатofункціональних блоків згідно з системою онтологічних патернів їх внутрішніх специфікацій.

Розроблення сучасних програмних засобів на основі стандартних онтологій дозволяє повторне спільне використання інформації відповідних предметних галузей як людьми, так і програмними агентами [4]. Використання онтологій з різних галузей уможливує повторне використання, аналіз і класифікацію знань. Обмін програмних агентів даними, описаними в термінах онтологій, дозволяє забезпечувати їхню взаємодію.

Подання знань в онтологічних моделях

Основною метою створення та використання онтологій як інформаційного ресурсу є забезпечення накопичення, розподілення та повторного використання знань в конкретній предметній галузі. Розробка онтологій вимагає уніфікації подання знань і мови маніпулювання ними. Стандарти форматів подання знань для комп'ютерної обробки розробляються та затверджуються консорціумом Всесвітньої павутини *W3C*, що дозволяє онтології відігравати роль загальної мови формального подання інформації, придатної для комп'ютерної обробки з урахуванням семантики [5].

Для подання знань прийнято абстрактну модель у вигляді *RDF*-триплету (*Resource Description Framework* – середовище опису ресурсу) «суб'єкт, предикат, об'єкт», що фактично є твердженням про ці сутності (елементи знань). Таке подання дає можливість декомпозиції знань та дозволяє задавати структуру опису джерела даних з урахування семантики. Суб'єкт, предикат і об'єкт є ресурсами з унікальним ідентифікатором *URI* (*Uniform Resource Identifier* – адресою ресурсу в інтернеті), що дає можливість однозначного визначення потрібної інформації. Стандарти на базі *RDF* дають можливість описувати

пов'язані з фактами логічні висновки, вказують, як знайти самі факти. Тобто задають спосіб зберігання та передачі онтологічних описів, метод обробки її концептів. Таке подання дає можливість поєднання декларативних та процедурних знань. Найбільш вживаними є формати *OWL/XML* та *RDFS* (*Resource Description Framework Schema*). Відповідно до цих стандартів було розроблено різні редактори онтологій.

Одним з найпопулярніших для роботи з онтологіями є редактор *Protege*, розроблений у Стенфордському університеті [6]. Він призначений для побудови та редагування онтологій предметної галузі на основі фреймів, відповідно до протоколу *OKBC* (*Open Knowledge Base Connectivity protocol* – прикладний інтерфейс програмування для доступу до баз знань систем подання знань). Побудовані моделі онтології складаються з множини класів у вигляді ієрархій, слотів, пов'язаних з класами для опису їхніх властивостей і відношень між ними, та екземплярів класів, які мають певні значення цих властивостей.

Protege є відкритою платформою з можливістю підтримувати більшість найпоширеніших форматів подання знань: *OWL*, *RDF/RDFS*, *SHOE*, *XOL*, *DAML+OIL*. Редактор також підтримує модулі розширення функціональності. В цій роботі прикладні результати отримано саме засобами цієї системи.

Структурні особливості комплексу інструментальних засобів індуктивного моделювання на основі МГУА

Станом на сьогодні відомо багато реалізацій методу групового урахування аргументів (МГУА [7, 8]), призначених для застосування в різних галузях: знаходження закономірностей, ідентифікація нелінійних систем, прогнозування стаціонарних і нестаціонарних процесів, управління складними технічними об'єктами тощо. При цьому кожен раз при

розв'язанні задачі виникають вимоги до конкретної практичної реалізації технології МГУА. Тому користувачам-дослідникам необхідно мати спектр доступних алгоритмів методу та їхніх складових для найбільш ефективного використання в рамках поточної задачі. Крім того, список алгоритмів методу постійно розширюється. Тож виникає нагальна потреба реалізації гнучкої та універсальної структури знань в галузі індуктивного моделювання, яка може бути використана для конструювання інструментальних засобів.

Основні вимоги до комплексу інструментальних засобів індуктивного моделювання:

1. Функціональність – можливість за даними спостережень отримати модель об'єкта з необхідними характеристиками та розв'язання практичних задач).

2. Адаптивність – можливість редагування та розширення функціональних можливостей за рахунок введення нових алгоритмічних модулів.

3. Підтримка діяльності з накопичення, розподілення та повторного використання знань з індуктивного моделювання.

4. Забезпечення підтримки семантичного супроводу користувача (тобто можливість отримати пояснення в процесі моделювання).

5. Універсальність – можливість використання як незалежної системи, так і елемента іншої системи.

Програмний комплекс інструментальних засобів індуктивного моделювання базується на таких основних компонентах:

- реалізація низки алгоритмів МГУА;
- інтерфейс користувача, який забезпечує взаємодію користувача з системою;
- БЗ у вигляді онтологічних моделей предметної галузі індуктивного моделювання, побудованих в онторедаторі *Protege*;
- засобів взаємодії БЗ та програмних модулів, реалізацій для отримання результату, потрібного користувачеві.

Інтерфейс користувача забезпечує взаємодію між користувачем та програмними компонентами інструментального комплек-

су, що означає забезпечення виконання низки завдань [9]:

- інтерпретація та інтеграція вхідної інформації з різних джерел на основі існуючої моделі даних;
- створення та коригування моделі діалогу;
- формування плану дій на основі моделі діалогу та минулих дій користувача (безпосереднє введення користувачем або відстеження його дій);
- подання результатів або формування нових запитів в процесі діалогу з користувачем.

Організація процесу індуктивного моделювання (ІМ) на основі МГУА в загальному випадку реалізується так. Після отримання вхідної вибірки даних виконується аналіз та за необхідності попередня обробка. Генерується задача: вибирається базис моделювання, метод генерації структур моделей, проводиться навчання і вибір кращих з них. Результати візуалізуються. Користувач за бажанням зберігає результати в БД, отримує звіт або продовжує моделювання, змінивши параметри (рис. 1). Етапи такого процесу моделювання, його складові частини визначаються відповідними онтологічними моделями (рис. 2 – 4).

Користувачу надається можливість розв'язання таких задач:

- 1) Отримання даних з різних джерел або їх генерація для тестування методів.
- 2) Вибір мети моделювання (побудова математичної моделі об'єкта; прогнозування процесу, заданого часовим рядом; виявлення інформативних ознак та відсів неінформативних; класифікація; класифікація тощо).
- 3) Визначення типу процесу (статичний об'єкт, часовий ряд, динамічний процес).
- 4) Вибір модельного класу (лінійна чи поліноміальна регресія, авторегресія, гармонічна, логарифмічна, поліноміальна, експоненціальна функції часу, різниці рівняння тощо).
- 5) Вибір зовнішнього критерію вибору моделі (Фішера, Акаїке, Маллоуза, критерій

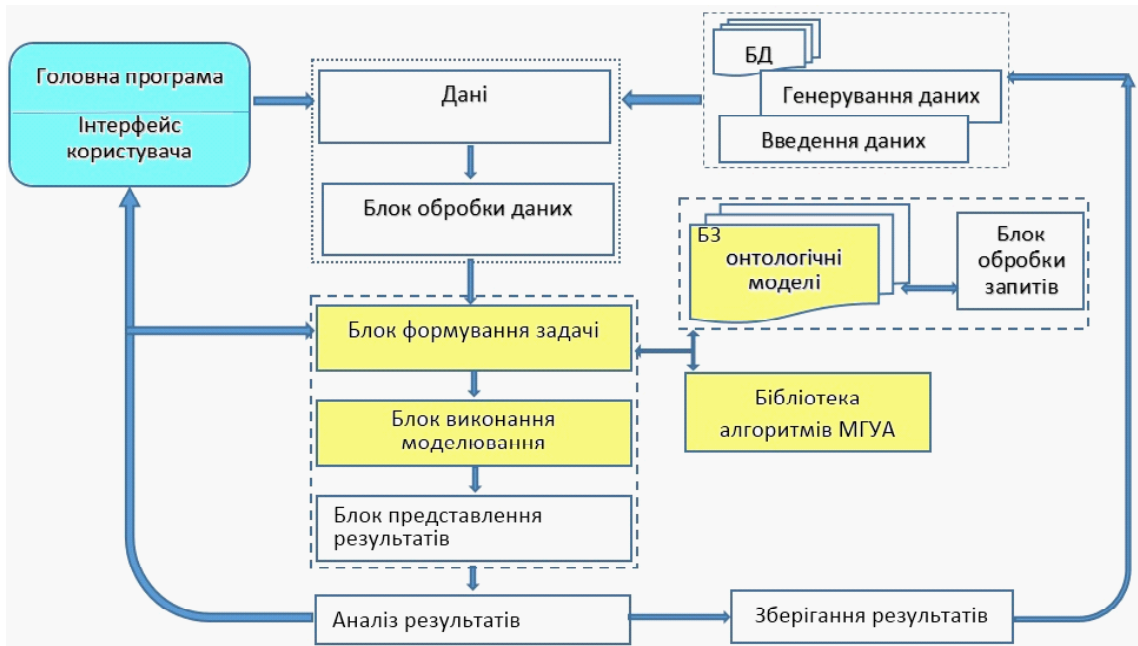


Рис. 1. Структурна схема комплексу інструментальних засобів ІМ з використання онтологій

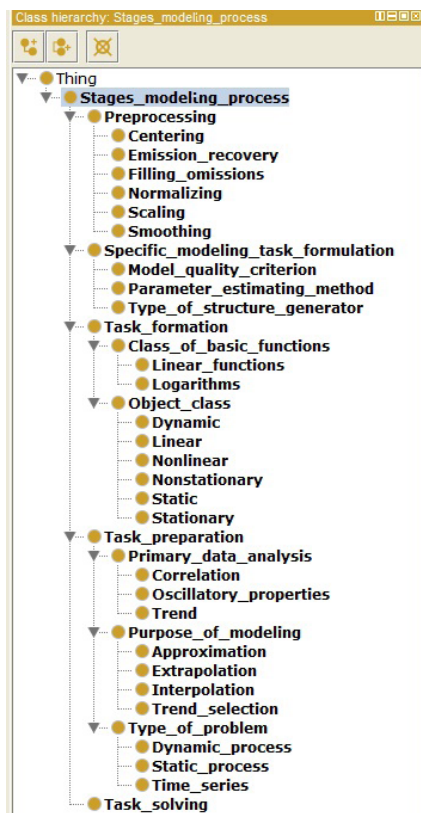


Рис. 2. Мета-онтологія процесу моделювання за статистичними даними

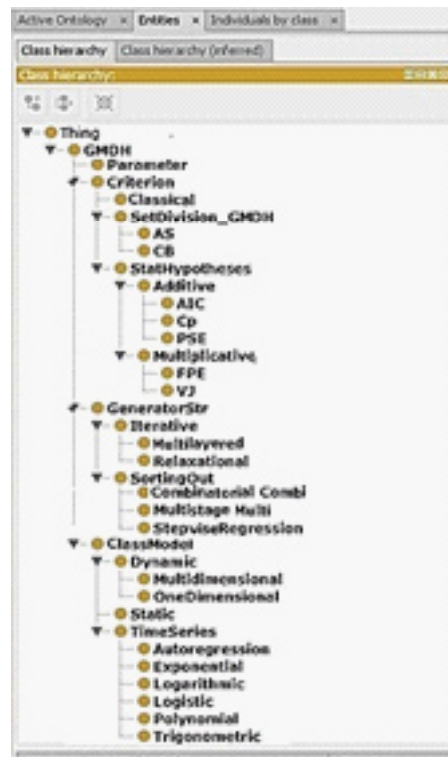


Рис. 3. Онтологічна модель алгоритмів МГУА

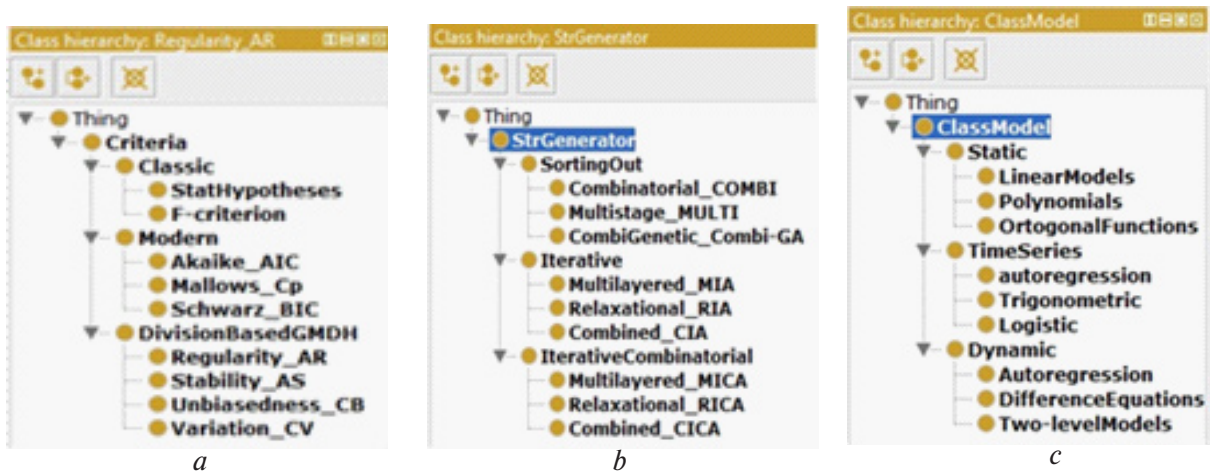


Рис. 4. Онтологічні моделі критеріїв вибору (а), генераторів структур моделей (б), класів базисних функцій (с)

регулярності, незміщеності, перехресне підтвердження (*cross-validation, jack-knife, leave-one-out*) тощо).

6) Вибір методу оцінювання параметрів (МНК, МНМ, гребенева регресія тощо).

7) Вибір методу генерації структур моделей (задана структура, комбінаторний, комбінаторно-селекційний, багаторядний, релаксаційний, генетичний, кореляційний тощо).

8) Перевірка адекватності моделі (статистика Фішера, критерій R^2 , точність на екзаменаційному наборі даних тощо).

Розглянемо структуру процесу індуктивного моделювання з точки зору формального означення онтології як триплету $O = \langle T, R, F \rangle$, де T – скінченна (непорожня) множина концептів (понять) предметної галузі, яку визначає онтологія; R – скінченна множина відношень між концептами предметної області, задає тип зв'язків; F – скінченна множина функцій інтерпретації, заданих на концептах та/або відношеннях онтології O .

Множина концептів онтології індуктивного моделювання містить означені поняття загальних класів (мета моделювання, тип процесу, клас моделі, критерій селекції,

метод оцінювання параметрів, генератор структур, алгоритм тощо) та відповідні їхні екземпляри.

Опис концептів задає різні типи властивостей, які складають множину відношень R та визначає тип зв'язків між концептами. Найбільш простими є відношення підпорядкованості «складається з», «є частиною», які задають ієрархію концептів. Крім них визначаються два типи властивостей. Об'єктні властивості (*Object-Property*) – зв'язки, які асоціюють об'єкти (класи), в якості значень можуть мати складні об'єкти. Властивості-характеристики (*DataProperty*) характеризують об'єкти (екземпляри) та приймають значення простих типів (*float, string, boolean, datatype, language, long* тощо)

Сукупність процедур, які розв'язують названі задачі та їх підзадачі, складає множину функцій інтерпретації F .

Інтелектуальність комп'ютерної системи передбачає, що певні етапи і відповідні подальші дії (в ідеальному випадку – весь процес моделювання) можуть бути виконані автоматично без участі користувача. Тобто, коли в залежності від результату виконання певної процедури система автоматично при-

ймає рішення про наступні дії згідно з логічними висновками, визначеними онтологією.

Наприклад, процедуру визначення типу процесу для онтологічного подання можна описати так:

Нехай m – кількість вхідних змінних.

Якщо $m=1 \Rightarrow \text{Тип_процесу} = \text{«часовий_ряд»}$.

Якщо $m>1 \vee (\text{«дані_вимірювались_у_часі»} \wedge \neg \text{Тип_процесу} = \text{«статичний_процес»}) \Rightarrow \Rightarrow \text{Тип_процесу} = \text{«динамічний_процес»}$.

Якщо $m>1 \vee (\neg \text{«дані_вимірювались_у_часі»} \wedge \neg (\text{Тип_процесу} = \text{«динамічний_процес»})) \Rightarrow \Rightarrow \text{Тип_процесу} = \text{«статичний_процес»}$.

Тут використано такі елементи множин онтології:

- Елементами множини T є концепти: *тип_процесу, часовий_ряд, статичний_процес, динамічний_процес, вхідні_змінні*;

- Елементами множини R є властивості: *мати_кількість_змінних, вимірюватись_у_часі*;

- Елементами множини функцій інтерпретації F є процедура визначення типу процесу.

Використання онтологій в БЗ інструментального комплексу

База знань є інформаційним ресурсом, який створюється інженером зі знань у вигляді онтологічних моделей предметної галузі індуктивного моделювання на основі МГУА у відповідних форматах *OWL/XML* та *RDFS*, побудованих засобами онторедатора *Protege_4.3*. Для структуризації доступних методів створюється мета-онтологія, яка виконує роль класифікатора. На рис. 2 наведено ієрархію загальних понять, які визначають основні етапи процесу побудови математичної моделі за статистичними даними. Вона задає ієрархію основних класів і може бути доповнена іншими онтологіями нижчого рівня, які деталізують предметну галузь та її складові. Так, онтологія на рис. 3 визначає основні складові алгоритмів МГУА. В свою чергу, на рис. 4 наведено онтологічні

структури, які описують множини класів базисних функцій, генераторів структур моделей, критеріїв, які застосовуються при побудові алгоритмів МГУА. Зв'язування онтологій здійснюється за допомогою унікального для кожного класу уніфікованого ідентифікатора ресурсів *URI*.

Онтологія складається з набору класів, організованих в ієрархію, для подання важливих понять домену, набору екземплярів (слотів), пов'язаних з класами, та їх властивостей. Для опису властивостей і відношень між класами та екземплярами задаються твердження загальної форми *ObjectProperty* (рис.5, *a*), які, в свою чергу, можуть бути складними об'єктами. Для атрибутів класів та екземплярів цих класів, які мають певні значення, визначаються твердження форми *DataProperty* (рис.5, *b*). При заданні властивостей вказується клас, до якого належить суб'єкт (область означення) та клас, до якого належить об'єкт (область значень). Для *ObjectProperty* суб'єктом і об'єктом є класи або екземпляри, які мають складну структуру. Для *DataProperty* область значень визначається одним з примітивних типів.

Подібним чином можна задавати і обмеження концептів. На рис. 6 наведено приклад обмеження на кількість вхідних даних для алгоритму повного перебору *Combi*, де визначається тип та значення атрибуту (фрагмент файлу подання онтології у форматі *OWL/XML*).

Реалізація макета комплексу інструментальних засобів індуктивного моделювання на основі МГУА

Інструментальний комплекс індуктивного моделювання складається з БЗ у вигляді онтологічних моделей предметної галузі, реалізацій алгоритмів МГУА та надбудови, яка дає можливість поєднати запити до БЗ та програми обчислень. На рис. 7 наведено

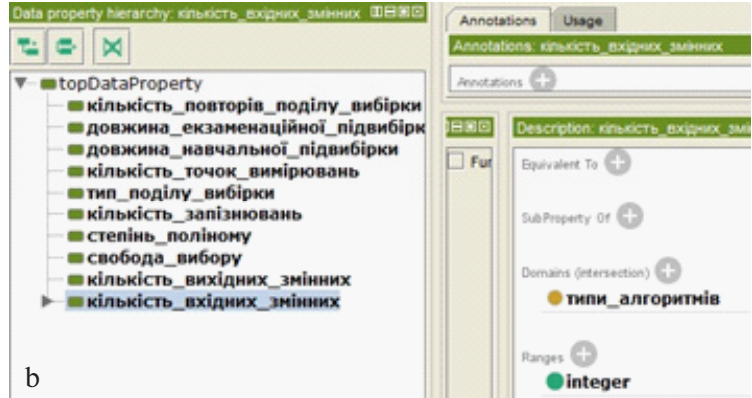
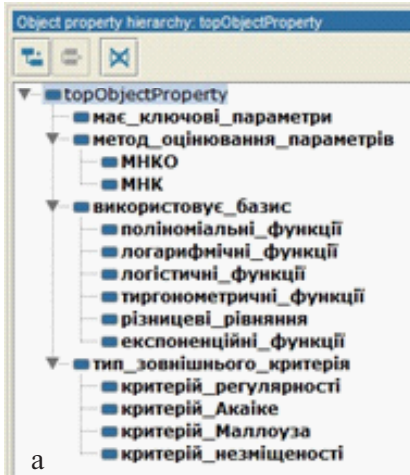


Рис. 5. Визначення відношень та атрибутів концептів онтологічної моделі предметної галузі ІМ засобами *Protege*

```
<Class IRI="#повний_перебір"/>
<DataMaxCardinality cardinality="30">
  <DataProperty IRI="#кількість_вихідних_змінних"/>
  <Datatype abbreviatedIRI="xsd:integer"/>
</DataMaxCardinality>
```

Рис. 6. Фрагмент онтологічного подання обмеження у форматі *OWL/XML*

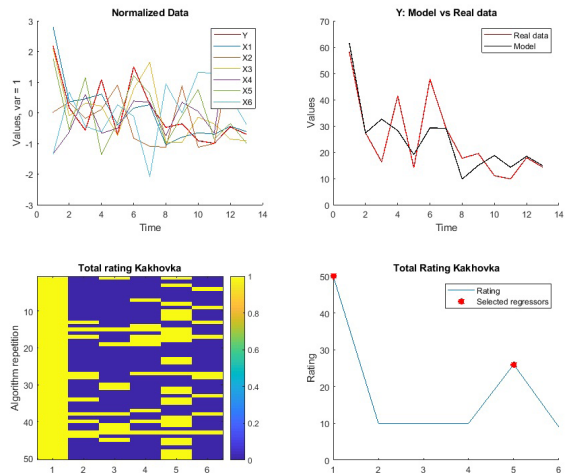


Рис. 8. Візуалізація результатів роботи алгоритму *CORAL*

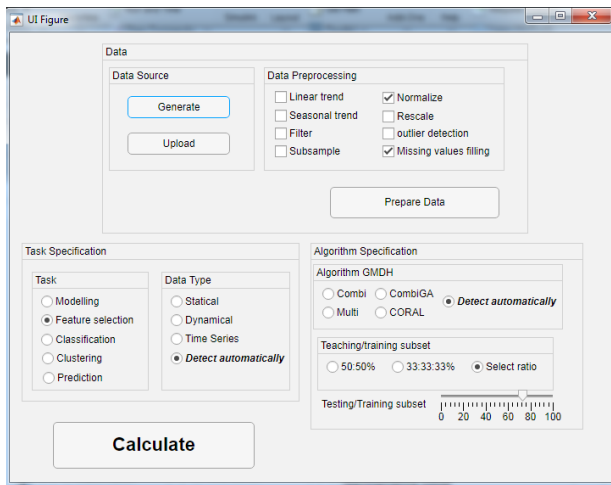


Рис. 7. Інтерфейс користувача макету інструментального комплексу ІМ

інтерфейс макета інструментального комплексу ІМ на базі МГУА. В комплексі передбачено можливість як явного визначення необхідних параметрів задачі користувачем, так і реалізації процедур автоматично-

го вибору алгоритму. Правила такого вибору задаються відповідними онтологічними моделями предметної галузі. Наприклад, при розв'язанні прикладної задачі виявлення найвагоміших фізико-гідрологічних чинників впливу на біопродуктивність фітопланктону в Каховському водосховищі [11] при застосуванні режиму автоматичного вибору алгоритму (рис. 7) буде обраний кореляційно-рейтинговий алгоритм *CORAL* шляхом обробки наявної інформації:

Стационарний об'єкт => клас поліноміальних моделей;

Кількість вхідних даних $<30, =>$ Перебірні алгоритми;

Мета моделювання – *Feature Selection* => *CORAL*.

На рис. 8 наведено приклад візуалізації результату роботи кореляційно-рейтингового алгоритму МГУА в задачі виявлення найвагоміших гідрофізичних чинників впливу на біопродуктивність фітопланктону в Каховському водосховищі [11].

Процедури обчислення (моделювання) реалізовано в середовищі *MatLabR2018b*.

Висновки

Розглянуто структуру комплексу інструментальних засобів індуктивного моделювання на основі МГУА. Особливістю є використання бази знань у вигляді онтології

предметної галузі індуктивного моделювання. Застосування онтологічного підходу до конструювання бази знань дає можливість повторного використання інформації та ефективного опрацювання цієї інформації в процесі моделювання складних об'єктів різної природи за статистичними даними, формувати запити й отримувати логічні висновки. Наведено фрагменти онтології індуктивного моделювання на основі МГУА як приклад створення формального опису предметної галузі. Для конструювання онтологічних моделей використано онторедактор *Protege_4.3*. Процедури обчислення, візуалізація результатів розв'язання практичної задачі та інтерфейс користувача макета інструментального комплексу реалізовано в середовищі *MatLabR2018b*.

ЛІТЕРАТУРА

1. Gruber Tom. 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal Human – Computer Studies*, 43(5–6), pp. 907–928
2. Model-driven development (MDD). 2018. <https://searchsoftwarequality.techtarget.com/definition/model-driven-development>
3. Dillon S., Chang E. *Ontology-Based Software Engineering – Software Engineering 2.0*. 2008. https://www.researchgate.net/publication/4328227_Ontology-Based_Software_Engineering-Software_Engineering_20. *Software Engineering*, 2008. ASWEC 2008. 19th Australian Conference. DOI: 10.1109/ASWEC.2008.4483185
4. Палагін, О.В., Петренко, М.Г. 2020. Знання-орієнтований інструментальний комплекс обробки баз даних наукових публікацій. *Системи керування та комп'ютери*, № 5. С. 17–33.
5. <https://www.w3.org/TR/owl2-overview>
6. <https://protege.stanford.edu/>
7. Stepashko V. 2019. On the Self-organizing Induction-Based Intelligent Modeling. *Advances in Intelligent Systems and Computing III / N. Shakhovska, M.O. Medykovskyy, Editors, AISC book series, vol. 871, Cham: Springer*. pp. 433–448.
8. Орлов А.А. 2013. Принципы построения архитектуры программной платформы для реализации алгоритмов метода группового учета аргументов. *УСиМ*, № 2. С. 65–71.
9. Pidnebesna, H., Pavlov A., Stepashko, V. 2020. *Ontology Application to Construct Inductive Modeling Tools with Intelligent Interface*. *Control Systems and Computers*, 4. С. 44–55.
10. Pidnebesna, H., Stepashko, V.: On Construction of Inductive Modeling Ontology as a Metamodel of the Subject Field. In: Proc. Int. Conf. “Advanced Computer Information Technologies”, University of South Bohemia, Ceske Budejovice, pp. 71–74 (2018).
11. Pidnebesna, H.A. 2020. Bioproductivity of Dnieper Reservoirs Analysis by Inductive Methods. *Control Systems and Computers*, № 2. С. 66–76.

Надійшла 23.11.2020

H. Pidnebesna, Junior Researcher,
International Research and Training Centre
for Information Technologies and Systems
of the NAS and MES of Ukraine,
03680, Kyiv, Glushkov ave., 40, Ukraine,
pidnebesna@irtc.org.ua

THE DESIGN OF INDUCTIVE MODELING TOOLS USING ONTOLOGIES

Introduction. The general trend in research of automation of designing of software systems is use of methods and means of ontological modeling both of design processes, and specifications of the developed systems and subject branches. Ontological modeling allows to shift the emphasis from the process of assembling software of application systems from large functional blocks to models of generating multifunctional blocks in accordance with the system of ontological patterns of their internal specifications.

The purpose of the work is to develop the principles of designing inductive modeling tools on the basis of ontological models of the subject area.

Methods - ontological models of the subject area in the construction of the instrumental complex of inductive modeling was developed using the onto editor Protege_4.3.

Result - The development of modern software based on standard ontologies allows the re-sharing of information in relevant subject areas by both humans and software agents. The use of ontologies from different fields allows the reuse, analysis and classification of knowledge. The exchange of software agents with the data described in terms of ontologies, allows to ensure their interaction.

Conclusions. The structure of the complex of tools of inductive modeling on the basis of GMDH is considered. A feature is the use of the knowledge base in the form of an ontology of the subject area of inductive modeling. The application of the ontological approach to the construction of the knowledge base makes it possible to reuse information and effectively process this information in the process of modeling complex objects of different nature according to statistical data, to form queries and obtain logical conclusions. Fragments of the ontology of inductive modeling based on GMDH as an example of creating a formal description of the domain are given. The Protege_4.3 onto editor was used to construct ontological models. The calculation procedures, visualization of the results of solving the practical problem and the user interface of the layout of the tool complex are implemented in the MatLabR2018b environmen.

Keywords: *inductive modeling, GMDH, ontological approach, intelligent systems, knowledge base.*