

АНОТАЦІЯ

Колонко М. Мова концептуального моделювання та методи для розробки баз даних зі багатоваріантною персистентністю. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії за спеціальністю 122 - Комп'ютерні науки. – Одеський національний політехнічний університет МОНУ, Україна, Одеса, 2020.

У *вступі* обґрунтовано актуальність вирішення завдання розробки розширення мови концептуального моделювання та методів автоматизації розробки баз даних зі багатоваріантною персистентністю з метою зниження часу проектування баз даних

У *першому розділі* дисертаційної роботи проаналізовано існуючі мови концептуального моделювання, технології персистентності, методи для автоматизації процесу розробки баз зі багатоваріантною персистентністю та обґрунтовано мету та задачі дослідження.

Показано, що і сьогодні процес проектування БД згідно класичної роботи Ramez Elmasri and Shamkant B. Navathe вимагає від ІТ-розробника вирішення низки задач при проведенні трьохетапного моделювання даних на концептуальному, логічному та фізичному рівні (Рис. 1).

На етапі концептуального проектування ІТ-розробники БД спільно зі спеціалістами предметної області (ПрО) створюють концептуальну модель, яка перетворюється в логічну модель. Проведено аналіз існуючих мов концептуального моделювання, а саме ER, ORM/NIAM, IDEF1X, EXPRESS, EER, UML, Information Analysis та AGILA MOD та показано що всі вони частково не відповідають вимогам, які висувають ІТ-розробники БД: є складними і не зрозумілими нетехнічним спеціалістам із ПрО для узгодження функціональних вимог, які висуваються до даних; не мають достатньої кількості засобів для автоматизованого відображення концептуальної схеми в логічну схему даних, щоб забезпечити мінімальне ручне втручання; орієнтовані на проектування тільки РБД і тому не використовуються при розробці NoSQL

або Polyglot Persistence БД. Показано що саме недосконалість існуючих мов концептуального моделювання призводить до підвищення витрат часу та трудомісткості проектування БД.

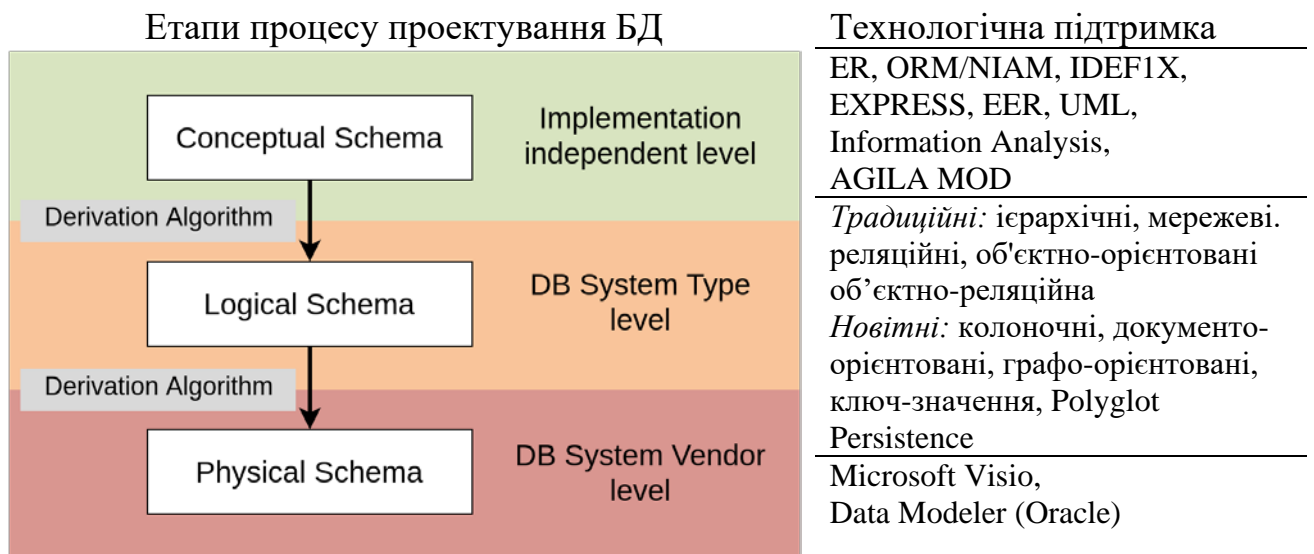


Рисунок 1 – Взаємозв'язок етапів проектування БД та їх технологічної підтримки

З точки зору можливостей відображення концептуальної схеми в логічну схему даних на логічному рівні проведено аналіз існуючих традиційних та новітніх логічних моделей даних. Показано спадкоємність структур з одного боку ієрархічних, мережевих та реляційних логічних моделей, а з іншого реляційних та колоночних; ієрархічних, мережевих та графо-орієнтованих; об'єктно-орієнтованих та документно-орієнтованих. Обґрунтовано переваги логічних моделей даних «ключ-значення» для створення пошукових БД.

Показано, що використанням різних технологій зберігання даних – Polyglot Persistence в рамках одного застосування на основі декількох баз даних з різними логічними моделями на стороні сервера має наступні переваги: хороша масштабованість застосування, ефективне управління різнорідними даними, більш швидкий час відгуку і як наслідок більш висока продуктивність. Але підводячи підсумки щодо процесу проектування БД на концептуальному та логічному рівні можна зробити висновок поширення баз даних NoSQL та концепція багатоваріантної персистентності зробили процес розробки бази даних із їх використанням ще більш довготривалим та трудомістким.

У зв'язку з перерахованим вище сформульовано мету і завдання дослідження.

У другому розділі дисертаційної роботи удосконалено мову концептуального моделювання AGILA MOD+ для автоматизації процесу розробки окремих концептуальних моделей даних зі багатоваріантною персистентністю. AGILA MOD (Automatic Generators for Information representation, Logical DB structures and Applications – AGILA Modeling language – MOD) – це автоматичний генератор для інформаційного представлення, логічних структур БД. Загальна ідея полягає в тому, щоб створити концептуальну модель шляхом перекладу природної мови, як речень, які семантично не спрощуються, в графічну нотацію. Основні структурні елементи є об'єкти O та асоціації A між ними показані на рисунку 2, а загальна концептуальна модель задається як (1).

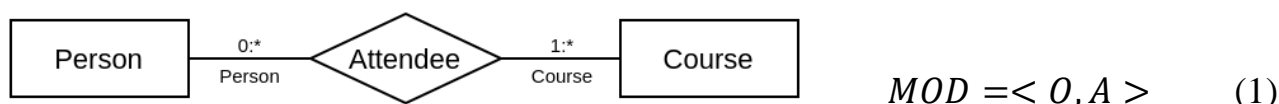


Рисунок 2 – Об'єкти-учасники асоціації

Введено формальний опис викладеного: нехай O є множина O об'єктів певних типів, потужність множини – $|O|$:

$$O = \{o(t)_1, o(t)_2, \dots, o(t)_i, \dots, o(t)_{|O|}\}, \quad (2)$$

Тип t об'єкта o_i задається як:

$$t = \left\{ \begin{array}{l} \text{inheritance} \{o_i \cap o_j \vee o_i \cup o_j\}, \text{abstract}, \\ \text{base} \{ \text{enumeration}, \text{set} \}, \\ \text{system} \{ \text{string}, \text{integer}, \text{boolean} \} \end{array} \right\} \quad (3)$$

де *system* – звичайний тип даних, наприклад «String», «Integer» або «Boolean»; *base* – базовий тип: перелік(*enumeration*) це фіксований список значень та набір (*set*) це впорядкована колекція без дублікатів; *inheritance* – задає успадкування між супертипом та його підтипами; *abstract* – складний тип об'єкта

Множину A асоціацій певних типів задано як:

$$A = \{a_1, a_1, \dots, a_i, \dots, a_{|A|}\} \quad (4)$$

де $|A|$ – потужність множини асоціацій

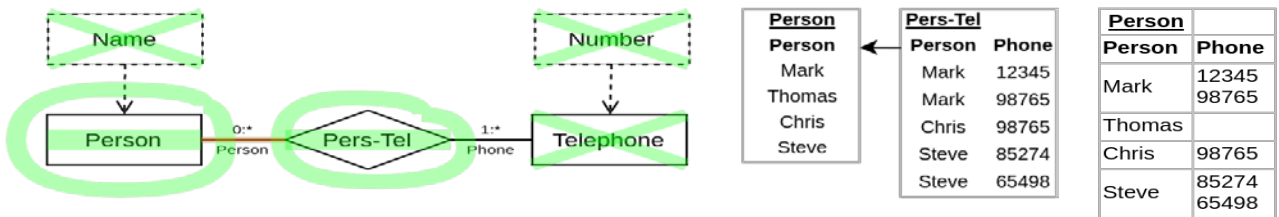
Тоді об'єкти o_i та o_j зв'язані за допомогою асоціації a_i , якщо:

$$\langle o_i, a_i, o_j \rangle = \{o_i \in O \wedge o_j \in O \wedge a_i \in A\} \quad (4)$$

Однією з найпотужніших, але простих конструкцій AGILA MOD є концепція *об'єктивізації асоціації*. Таким чином, створюється структурований об'єкт. Об'єктивізація асоціації дозволяє розробнику моделі повторно використовувати асоціацію як об'єкт для моделювання подальших речень, яке семантично не спрощується. Об'єктивізація асоціації має вигляд:

$$\langle (o_i, a_i, o_j) \vee ((o_i, a_i, o_j), a_j, o_k) \rangle = \{o_i, o_j, o_k \in O \wedge (o_i, a_i, o_j) \in O \wedge a_i, a_j \in A\} \quad (5)$$

На прикладі концептуальної моделі телефонної книги (рис. 2,а) побудованої за допомогою існуючих структурних елементів AGILA MOD розглянуті можливості фреймворку AGILA щодо автоматизованої деривації логічних моделей даних (рис. 3, b,c).



(a) Концептуальна модель із знаками деривації

(b) Реляційна модель

(c) Об'єктно-реляційна модель

Рисунок 3 – Приклади можливої та неможливої деривації концептуальної моделі

Показано, існуючий синтаксис мови AGILA MOD має недоліки щодо визначення та відображення автономних (Standalone) об'єктів, каскадного видалення та оновлення (Cascading), а також вкладених структури (Nested Structures). З врахуванням знайдених недоліків запропоновано удосконалити синтаксис мови AGILA MOD (структуру концептуальної моделі) за рахунок введення двох структурних елементів: незалежний об'єкт (*autonomous*) та «сильного» типу зв'язку об'єкта-учасника асоціації (Strong Association Participants)

Незалежний об'єкт виражає, що екземпляри описаного елемента Про існують незалежно від його відношень іншими елементами, позначається

прямокутником із вертикальними подвійними лініями (Рис.3,а). Його введення дозволяє скорегувати існуюче в фреймворку AGILA правило про необхідність скасування об'єктів-учасником асоціації з нижньою межею «1» та які не є частиною компонента узгодженості. Отримана реляційна модель (рис. 4,b) на відміну моделі (рис. 3, b) показує, що тепер незалежний об'єкт «Telephone» отримує своє власне відношення, на яке посилається відношення «Pers-Tel»

Сильний тип зв'язку об'єкта-учасника асоціації означає що асоціація сильніше пов'язана із підключеним об'єктом з межами участі в асоціації «x: 1» ніж з рештою. Тобто в асоціації «Pers-Tel» «Person» володіє «Telephone». У AGILA MOD + такий об'єкт-учасник підключається до асоціації за допомогою товстої лінії (рис 5,а). Для деривації із концептуальної моделі логічної моделі в фреймворку AGILA використовують наступний сценарій:

1. один тип об'єкта всередині компонента узгодженості позначається як ім'я відношення;

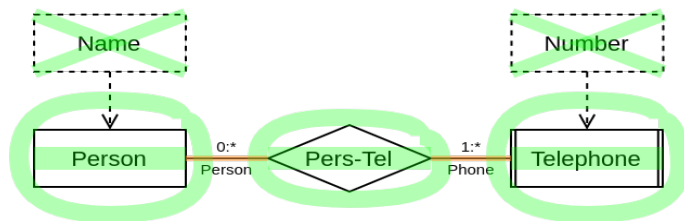
2. усі інші типи об'єктів усередині відношення стають ідентифікуючими атрибутами;

3. усі ролі типу асоціації, що лежать усередині компонента узгодженості, які вказують на тип об'єкта поза компонентом узгодженості, стають атрибутом відношення;

4. ролі цього типу, які є результатом позначеної сильної участі з верхньою межею, більшою за 1, інкапсулюються як вкладені атрибути.

При цьому пункт 4 зазначеного сценарію виконується тільки в разі деривації концептуальної моделі (рис. 5,а) в об'єктно-реляційну модель даних на логічному рівні (рис. 5,b).

Таким чином, виконане друге завдання дисертаційного дослідження та отримано перший пункт наукової новизни, а саме удосконалено структуру концептуальної моделі даних за рахунок розробки незалежного типу об'єктів та введення «сильного» типу зв'язку для об'єкта-учасника асоціації, що дало можливість вдосконалити правила деривації логічних моделей з урахуванням багатоваріантної персистентності даних.

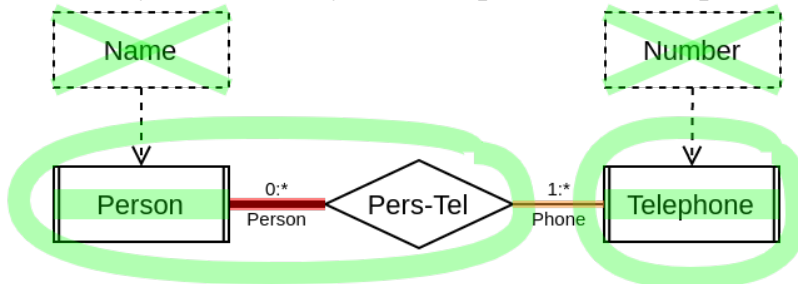


(a) Концептуальна модель із знаками деривації, які враховують незалежний об'єкт

Person	Pers-Tel	Telephone
Mark	Mark 12345	12345
Thomas	Mark 98765	98765
Chris	Chris 98765	85274
Steve	Steve 85274	65498
Steve	Steve 65498	

(b) Реляційна модель

Рисунок 4 – Результат деривації з використанням незалежного об'єкта.



(a) Концептуальна модель із знаками деривації, які враховують незалежні об'єкти та сильний тип зв'язку учасників асоціації

Person	Phone	Telephone
Mark	12345	12345
Mark	98765	98765
Thomas		85274
Chris	98765	65498
Steve	85274	
Steve	65498	

(b) Об'єктно-реляційна модель

Рисунок 5 – Результат деривації з використанням незалежного об'єкта та сильної участі в асоціації

У третьому розділі дисертаційної роботи створено методи автоматизації розробки логічних моделей баз даних на основі концептуальних субмоделей зі багатоваріантною персистентністю.

Адаптований до класичного (див. рис. 1) процес проектування БД представлено наступним чином (рис.6):

1. Створення комплексної концептуальної моделі Про відбувається за допомогою засобів розширеної мови AGILA MOD+. Далі виконується – розділення комплексної концептуальної моделі на незалежні концептуальні субмоделі, які враховують особливості персистентності даних на логічному рівні.

2. Послідовне перетворення за допомогою алгоритму деривації отриманих концептуальних субмоделей у відповідні дано логічні моделі – «логічні схеми», вигляд яких залежить від типу персистентності даних, який підтримується засобами конкретної СУБД.

3. Послідовне перетворення відповідних дано логічних моделей у відповідні фізичні моделі (схеми) – внутрішні структури персистентності конкретної СУБД.

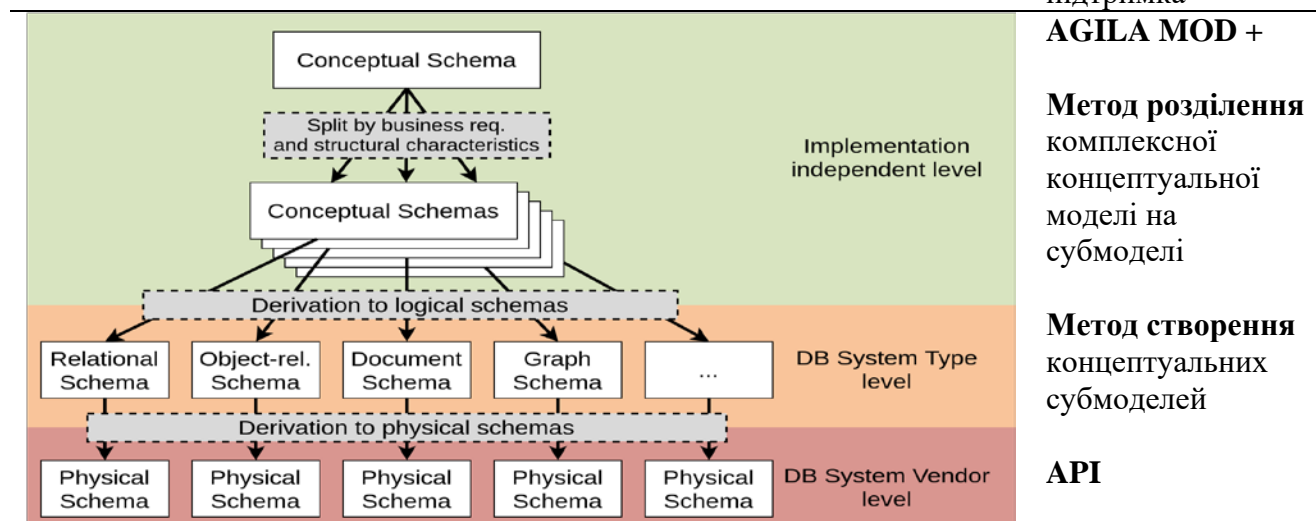


Рисунок 6 – Взаємозв’язок етапів адаптованого процесу проектування БД з урахуванням багатоваріантної персистентності та розробленої їх методичної підтримки

Для методичної підтримки дій першого етапу розроблено метод розділення комплексної концептуальної моделі ПрО на незалежні концептуальні субмоделі для подальшої їх деривації у відповідні логічні моделі. Метод включає такі етапи:

1. Створення копії існуючої комплексної концептуальної моделі.
2. Оточення замкненою, суцільною напівпрозорою оболонкою елементів комплексної концептуальної моделі об’єктів, асоціацій та ліній зв’язків між ними, із яких створюється субмодель.
3. Видалення усіх асоціацій та підключених до них ліній зв’язків, що лежать зовні оболонки оточення.
4. Видалення усіх об’єктів системних та базових типів, що лежать зовні оболонки оточення, і не мають з’єднання з об’єктами, які належать до внутрішньої частини оточення.
5. Залишення усіх об’єктів системних та базових типів, що лежать зовні оболонки оточення, та мають з’єднання з об’єктами що належать до внутрішньої частини оточення.
6. Заміна типу усіх залишених об’єктів, що лежать зовні оболонки оточення і мають з’єднання всередині на тип «посилання» (referencing) на субмодель, до якої належить оригінальний об’єкт.

Таким чином, в результаті розділення комплексної концептуальної моделі маємо три множини елементів (об'єктів, асоціацій та ліній зв'язків між ними): «inside», «outside» та «remainder». До множини «inside» належать елементи оточені оболонкою на другому кроці, до «outside» – елементи які було видалено на третьому та четвертому кроці. Множина «remainder» складається із об'єктів що мають тип «посилання», лежать зовні оболонки оточення, і мають з'єднання з об'єктами, що належать множини «inside».

Для побудови оболонки – оточення елементів комплексної концептуальної моделі запропоновано метод створення концептуальних субмоделей, які враховують особливості персистентності даних. Метод передбачає проведення аналізу структурних особливостей множини елементів комплексної концептуальної моделі, а саме об'єктів, асоціацій та ліній зв'язків між ними:

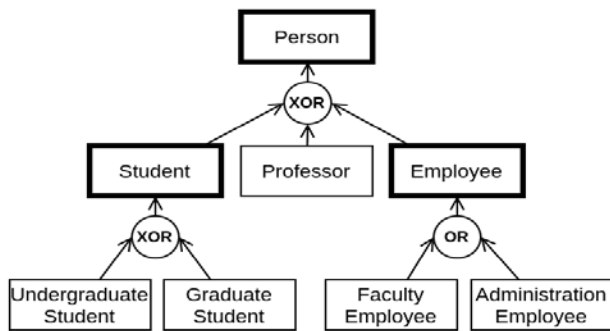
1. Наявність сильних зв'язків (див. рис. 5,а) між об'єктом та асоціацією, які запропоновані в AGILA MOD+, є показником віднесення цих елементів до документо-орієнтованих або об'єктно-реляційних структур персистентності

2. Наявність ієрархії типів об'єктів є показником віднесення цих елементів до документо-орієнтованих або об'єктно-реляційних структур персистентності. За допомогою ієрархії типів об'єктів AGILA MOD підтримує успадкування об'єктів. На рисунку 7 а, б та с показані концептуальна модель і результати деривації успадкування.

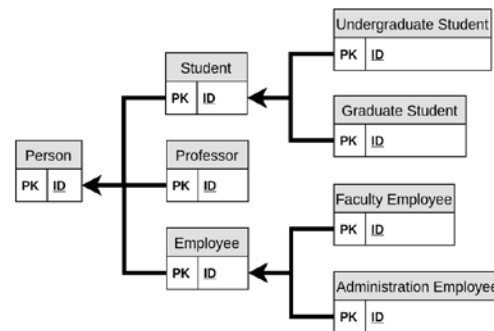
3. Для концептуальних моделей або їх частин, що містять значну кількість успадкувань, особливо коли успадкування є складним із використанням довільних комбінацій “OR”, “XOR” та “AND” доцільно використовувати документно-орієнтовані структури персистентності XML або JSON.

4. Наявність великої кількості асоціацій типу «Самоасоціація» (Self Association) коли межі зв'язків об'єктів-учасників дорівнюють "0: *" є

показником віднесення цих елементів до графо-орієнтованих структур персистентності



(a) концептуальна модель ієрархії типів об'єктів



(b) реляційна модель ієрархії типів об'єктів

```

<s:complexType name="studentType"
  abstract="true">
  <s:complexContent>
    <s:extension base="personType">
      <s:sequence>
        <s:element
          name="studyProgram" minOccurs="1"
          type="s:string"
        <s:element
          name="semester"
  
```

```

type="s:positiveInteger" maxOccurs="1" />
  <s:element name="enrolmentNo"
type="s:positiveInteger"
  </s:sequence>
</s:extension>
</s:complexType>
  
```

(c) документо-орієнтована модель (XML)

Рисунок 7 – Модель та результати деривації успадкування

Апробація розроблених методів на прикладі проекту KampfrichterAnmelde-System (KRAS) – веб-додатку, який було створено для інформаційної підтримки Баварської Федерації дзюдо – дозволила розділити існуючу комплексну концептуальну модель ПрО на дві незалежні субмоделі для подальшої їх деривації в реляційну та LDAP моделі логічного рівня та створення відповідних БД. Дослідження показали, що було отримано скорочення в 2,3 рази часу на таке проектування двох БД в порівнянні з їх розробкою за класичною схемою.

Таким чином отримані другий та третій пункти наукової новизни а саме:

- вперше розроблено метод розділення комплексної концептуальної моделі даних на незалежні субмоделі що дозволило автоматизувати отримання відповідних логічних моделей даних включаючи реляційні та NoSQL структури та тим самим скоротити час проектування баз даних;

– вперше розроблено метод створення концептуальних субмоделей, які враховують особливості персистентності даних включаючи реляційні та NoSQL структури із комплексної концептуальної моделі даних, що дозволило обґрунтовано вибрати тип структур даних на логічному рівні і тим самим зменшити час розробки баз даних з урахуванням багатоваріантної персистентності.

Врахування удосконаленої структури концептуальної моделі даних та методів автоматизації розробки логічних моделей баз даних дозволило отримати четвертий пункт наукової новизни:

– удосконалено мову концептуального моделювання AGILA MOD+ за рахунок за рахунок удосконаленої структури концептуальної моделі даних та методів створення моделей даних концептуальному рівні, що дозволило враховувати багатоваріантну персистентність та скоротити час проектування баз даних інформаційних систем.

У *четвертому розділі* розроблено інтерфейс прикладного програмування (API) для забезпечення єдиного доступу до баз даних зі багатоваріантною персистентністю із бізнес-логіки

У *п'ятому розділі* проведено апробацію мови концептуального моделювання AGILA MOD+ та API при розробці БД в процесі виконання ІТ-проектів та показано суттєве зменшення часу проектування, що підтверджено відповідними актами впровадження.

Ключові слова: Розробка баз даних, багатоваріантна персистентність, моделювання предметних областей (PrO), мови концептуального моделювання, модель сутність-зв'язок, NoSQL бази даних, реляційні бази даних, система управління базами даних (СУБД).

ABSTRACT

Kolonko M. Conceptual modeling language and methods for developing databases with polyglot persistence . - Qualifying scientific work on the manuscript.

Thesis for the PhD degree in specialty 122 «Computer science» (12 - Information Technology). - Odessa National Polytechnic University, Ukraine, Odessa, 2020.

The *introduction* verifies the topicality of solving the problem of developing an extension of the conceptual modelling language and methods for automating the development of databases with multivariate persistence in order to reduce the time of database design.

In the *first chapter* of the thesis paper, the existing languages of conceptual modelling, persistence technologies, methods for automating the process of developing databases with multivariate persistence are analyzed and the goal and objectives of the research are justified.

It is shown that today the process of designing a database according to the classic work of Ramez Elmasri and Shamkant B. Navathe requires IT-developer to solve a number of problems when carrying out a three-stage data modelling at the conceptual, logical and physical levels (Fig. 1).

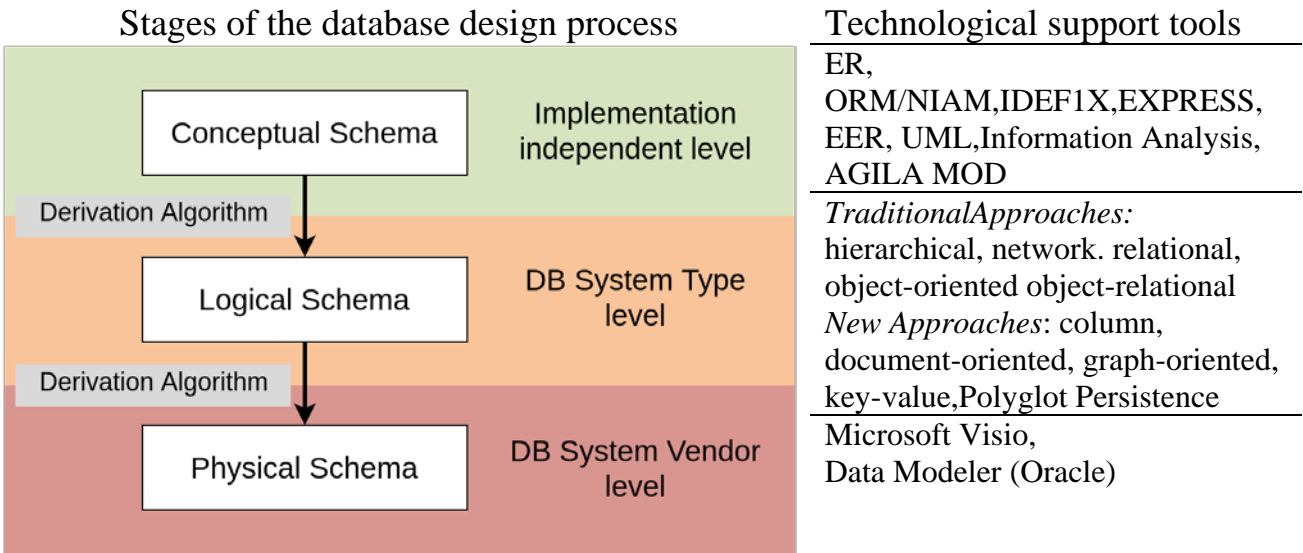


Figure 1 – Relationship between the stages of database design and their technological support

At the conceptual design stage, IT database developers, together with the Universe of Discourse (UoD) experts, create a conceptual model which is further converted into a logical one. The capabilities of the existing conceptual modelling languages have been analyzed, namely: ER, ORM / NIAM, IDEF1X, EXPRESS, EER, UML, Information Analysis and AGILA MOD, and it is shown that they all have the following disadvantages: they are partially inconsistent with the requirements of IT database developers; they are complex and incomprehensible to non-technical experts of the subject domain who are involved in reconciling the

functional requirements for the data; they do not have sufficient tools to automatically transform a conceptual schema into a data logic one with minimal manual intervention; they are focused on designing only relational databases and therefore are not used in the development of NoSQL or Polyglot Persistence databases. It is shown that it's the imperfection of the existing languages of conceptual modelling that leads to an increase in the time and manpower input (labour intensity) when designing the database.

From the point of view of the possibilities of converting a conceptual schema into a logical data schema, at the logical level the analysis of existing traditional and newest logical data models has been carried out. The continuity of traditional logical data models, such as hierarchical, network and relational ones, is shown. As well as within the study the continuity of traditional and new logical models has been also shown, for example, the relationship between relational and column-family models; between inter-hierarchical, network and graph-oriented; between object-oriented and document-oriented ones. The advantages of "key-value" logical data models for the creation of search databases have been substantiated.

It is shown that the use of different data storage technologies (Polyglot Persistence) within one application based on several databases with different logical models on the server-side has the following advantages: good scalability of the application, efficient management of heterogeneous data, faster response time and, as a result, higher performance.

But summing up what has been said about the process of designing a database at the conceptual and logical level, the author can conclude that the spread of NoSQL databases and the concept of multivariate persistence made the process of developing a database with their use even more time- and labour-consuming.

In connection with the above, the first section also formulates the purpose and objectives of the study.

The *second chapter* of the dissertation the language of conceptual modeling AGILA MOD + is improved for development of process automation of separate data conceptual models with multivariate persistence. AGILA MOD (Automatic

Generators for Information Representation, Logical DB structures and Applications - AGILA Modeling language - MOD) is an automatic generator for information representation, logical structures of databases. The general idea is to create a conceptual model by translating natural language as sentences that are not semantically simplified into graphic notation. The main structural elements are the objects O and the associations A between them are shown in Figure 1, and the general conceptual model is given as (1).

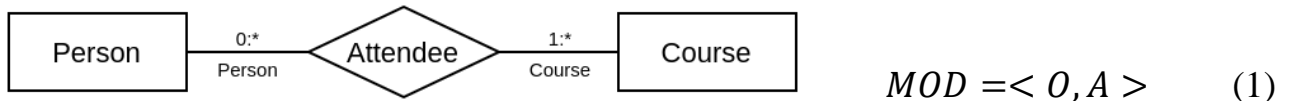


Figure 2 – Objects-members of the association

The formal description is introduced: let there be a set O of objects of certain types, cardinality of set – $|O|$:

$$O = \{o(t)_1, o(t)_2, \dots, o(t)_i, \dots, o(t)_{|O|}\}, \quad (2)$$

Type t of object o_i is defined as:

$$t = \left\{ \begin{array}{l} \text{inheritance} \{o_i \cap o_j \vee o_i \cup o_j\}, \text{abstract}, \\ \text{base} \{ \text{enumeration, set} \}, \\ \text{system} \{ \text{string, integer, boolean} \} \end{array} \right\} \quad (3)$$

where *system* – the usual data type, for example «String», «Integer» або «Boolean»; *base* – basic type: enumeration is a fixed list of values and a set is an ordered collection without duplicates; *inheritance* – specifies the inheritance between the supertype and its subtypes; *abstract* – complex object type

A set A of associations of certain types is given as:

$$A = \{a_1, a_1, \dots, a_i, \dots, a_{|A|}\} \quad (4)$$

where $|A|$ – power set of associations

Then objects o_i and o_j are connected by association a_i , if:

$$\langle o_i, a_i, o_j \rangle = \{o_i \in O \wedge o_j \in O \wedge a_i \in A\} \quad (4)$$

One of the most powerful but simple constructions of AGILA MOD is the concept of *objectification association*. Thus, the structured object is created. Objectification association allows the model developer to reuse the association as an object to model further sentences, which is not semantically simplified. The objectification of the association has the form:

$$\langle (o_i, a_i, o_j) \vee ((o_i, a_i, o_j), a_j, o_k) \rangle = \{o_i, o_j, o_k \in O \wedge (o_i, a_i, o_j) \in O \wedge a_i, a_j \in A\} (5)$$

Example of a conceptual model of a telephone book (Fig. 3, a) built with the help of existing structural elements of AGILA MOD shows the possibilities of AGILA framework for automated derivation of logical data models (Fig. 3, b, c).

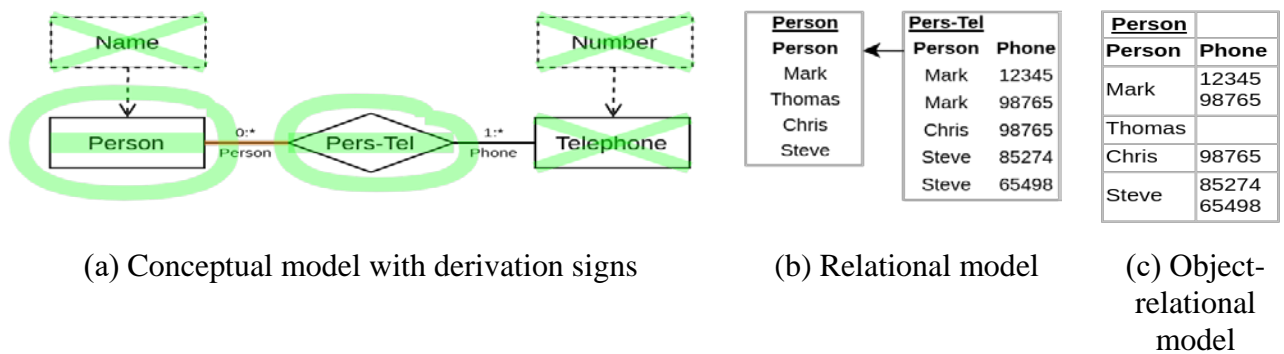


Figure 3 – Examples of possible and impossible derivation of a conceptual model

It is shown that the existing syntax of AGILA MOD language has disadvantages in terms of defining and displaying Standalone objects, Cascading, and Nested Structures. Taking into account the disadvantages found, it is proposed to improve the syntax of AGILA MOD language (structure of the conceptual model) by introducing two structural elements: an *autonomous object* and *Strong Association Participants*.

An autonomous object expresses that the instances of the described element of UoD exist independently of its relations with other elements, is denoted by a rectangle with vertical double lines (Fig. 4, a). Its introduction allows to adjust the existing in AGILA framework about the need to cancel the participating objects of the type of association "with the lower limit "1" and is not part of the consistency component. The obtained relational model (Fig. 4, b) in contrast to the model (Fig. 3, b) shows that now An autonomous object "Telephone" receives its own relation, which is referred to by the relation "Pers-Tel".

Strong Association Participants means that the association is strongly associated with the connected object with «x: 1» than with the rest. That is, in the association "Pers-Tel" "Person" owns "Telephone". In AGILA MOD +, such strong participation in association using a thick line (Figure 5, a).

To derive from the conceptual model of the logical model into the AGILA framework, the following scenario is used:

1. one object type within a consistency component is referred to as a name;
2. all other object types within the relationship become identifying attributes;
3. all association type roles that lie inside the consistency component that point to an object type outside the consistency component become an attribute of the relationship;
4. roles of this type, which are the result of marked strong participation with an upper limit bigger than 1, are encapsulated as nested attributes

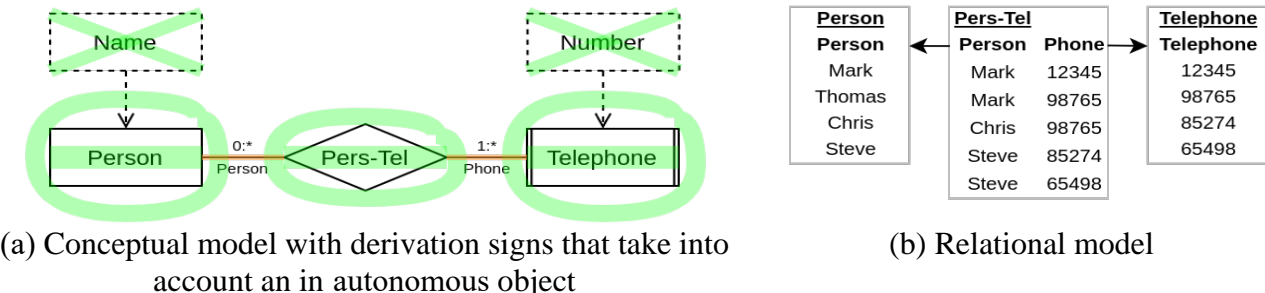


Figure 4 – The result of derivation using an autonomous object

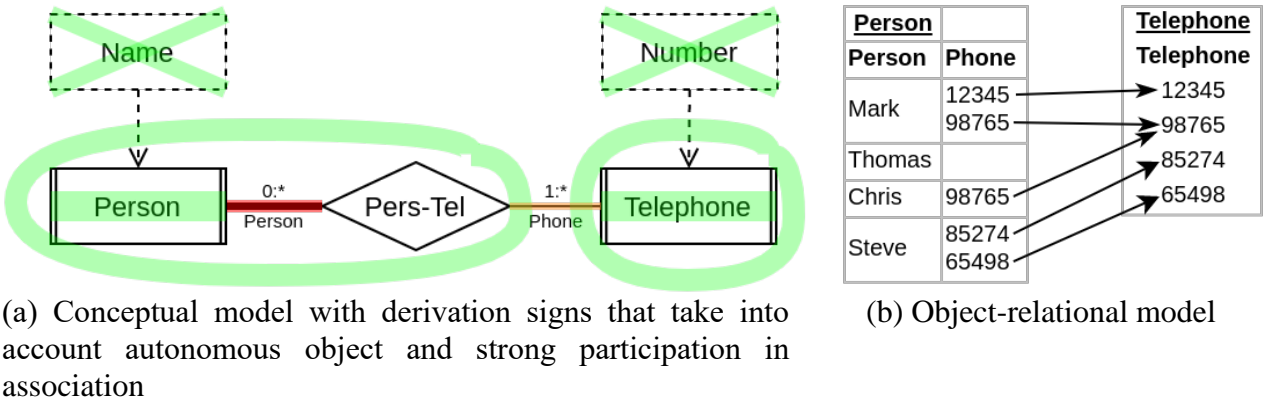


Figure 5 – The result of derivation using an autonomous object and strong participation in association

In this case, paragraph 4 of this scenario is performed only in case of derivation of the conceptual model (Fig. 5, a) in the object-relational data model at the logical level (Fig. 5, b).

Thus, the second task of the dissertation research was completed and the first point of scientific novelty was obtained, namely, the structure of the conceptual data model has been improved through the development of an independent type of objects and the introduction of a Strong Association Participants, which made it possible to

improve logical model derivation rules taking into account the Polyglot persistence of data.

The *third chapter* of the dissertation represents methods of automation of database logical models development on the basis of conceptual submodels with multivariate persistence.

Adapted to the classical (see Fig. 1) database design process is represented as (Fig. 6):

1. Creation of a comprehensive conceptual model of UoD is with the help of extended language AGILA MOD +. Then an additional step is performed - division of a complex conceptual model into independent conceptual submodels, which take into account the features of data persistence at the logical level.

2. Sequential transformation using the algorithm of derivation of the obtained conceptual submodels into appropriate pathological models - "logical schemes", the appearance of which depends on the type of data persistence, which is supported by a specific database.

3. The corresponding data models are consistently transformed into the corresponding physical models (schemes) - internal structures of persistence of concrete DBMS.

To methodically support the actions of the first stage, a method of dividing the complex conceptual model of ABM into independent parts, the so-called "sections" of the conceptual model for their further derivation into appropriate logical models, was developed. The method includes the following stages:

1. Creating a copy of the existing complex conceptual model.
2. Surrounding with a closed, continuous translucent shell of elements of the complex conceptual model of objects, associations and lines of connections between them, from which the submodel is created.
3. Delete all associations and connected lines that lie outside the shell.
4. Delete all system and base type objects that lie outside the environment shell and have no connection to objects that belong to the inner part of the environment.

5. Abandoning of all objects of the system and basic types that lie outside the shell of the environment and have connections with objects that belong to the inner part of the environment.

6. Replacement of the type of all abandoned objects that lie outside the shell of the environment and have a connection inside the type of "reference" (referencing) to the submodel to which the original object belongs.

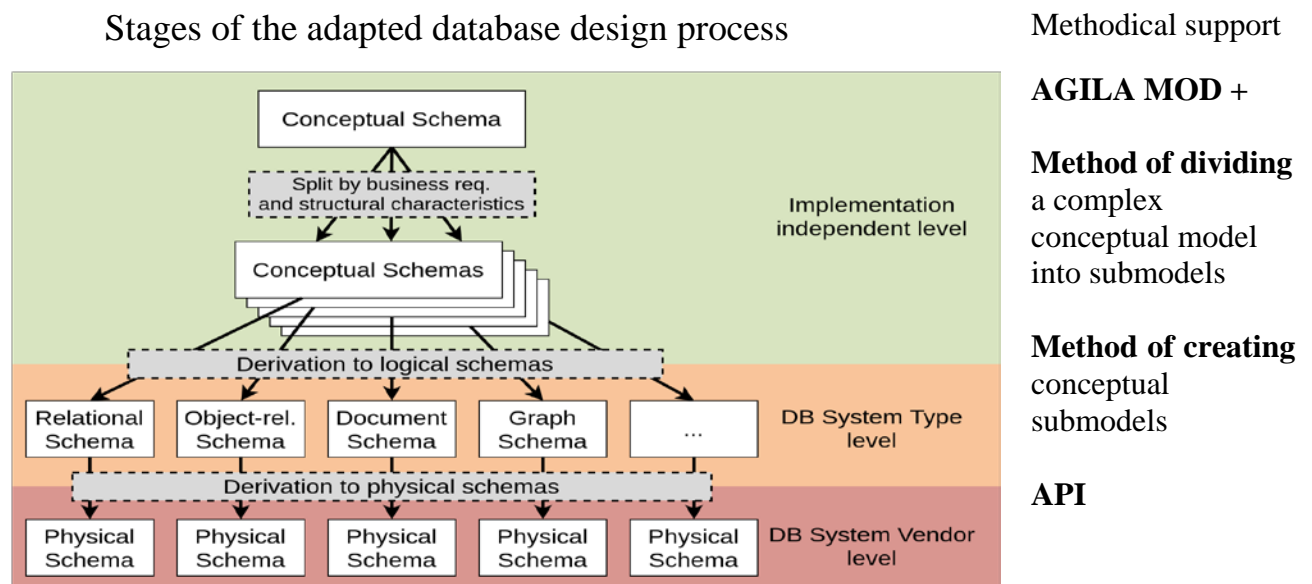


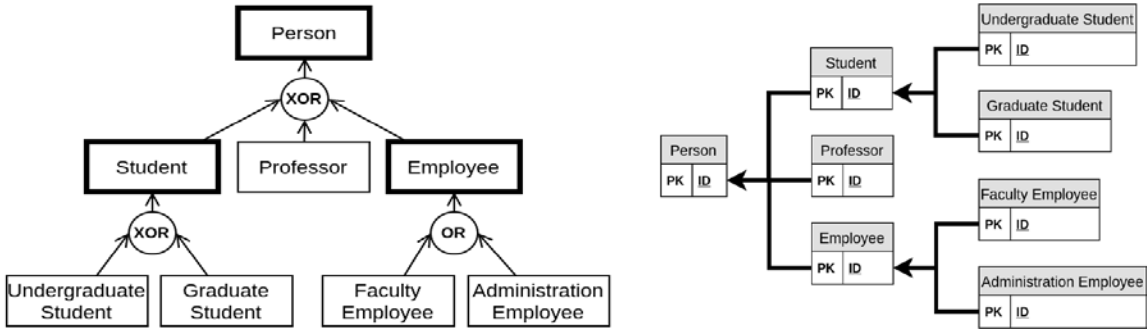
Figure 6 – The interconnection of the stages of the adapted process of database design, taking into account the Polyglot persistence and their developed methodological support

Thus, as a result of the division of a complex conceptual model, we have three sets of elements (objects, associations and lines of connection between them): «inside», «outside» and «remainder». The set «inside» includes elements surrounded by a shell in the second step, the «outside» - elements that were removed in the third and fourth steps. The set «remainder» consists of objects that have the type «link», lie outside the shell of the environment, and have connections with objects belonging to the set «inside».

To build shell – the environment of the complex conceptual model elements, a method of creating conceptual submodels that take into account the peculiarities of data persistence is proposed. The method involves the analysis of structural features of many elements of a complex conceptual model, namely objects, associations and lines of connection between them:

1. The presence of strong links (see Fig. 5, a) between the object and the association, which are proposed in AGILA MOD +, is an indicator of the attribution of these elements to document-oriented or object-relational structures of persistence.

2. The presence of a hierarchy of object types is an indicator of the attribution of these elements to document-oriented or object-relational structures of persistence. AGILA MOD supports object inheritance through a hierarchy of object types. Figure 7 a, b and c show the conceptual model and results of inheritance derivation.



(a) conceptual model of object type hierarchy

(b) relational model of the hierarchy of object types

<pre> <s:complexType name="studentType" abstract="true"> <s:complexContent> <s:extension base="personType"> <s:sequence> <s:element name="studyProgram" minOccurs="1" type="s:string" <s:element name="semester" </pre>	<pre> type="s:positiveInteger" maxOccurs="1" /> <s:element name="enrolmentNo" type="s:positiveInteger" </s:sequence> </s:extension> </s:complexContent> </s:complexType> </pre>
---	---

(c) document-oriented model (XML) inheritance output

Figure 7 – Model and results of inheritance derivation

For conceptual models or their parts that contain a significant number of inheritances, especially when inheritance is complex using arbitrary combinations of “OR”, “XOR” and “AND”, it is better to use document-oriented XML or JSON persistence structures.

3. The presence of a large number of associations such as "Self-Association" when the boundaries of the links of the objects-participants are "0: *" is

an indicator of the attribution of these elements to graph-oriented structures of persistence.

Approbation of the developed methods on the example of the project KampfrichterAnmelde-System (KRAS) - a web application created for information support of the Bavarian Judo Federation - allowed to divide the existing complex conceptual model of UoD into two independent submodels for their further derivation into relational and LDAP models and creation of appropriate databases. Studies have shown that there was a reduction of 2.3 times the time for such design of two databases compared to their development according to the classical scheme.

In this way obtained the second and third points of scientific novelty, namely:

- for the first time a method of dividing a complex conceptual data model into independent submodels was developed, which allowed to automate the production of appropriate logical data models including relational and NoSQL structures and thus reduce database design time;

- for the first time a method of creating conceptual submodels that take into account the peculiarities of data persistence, including relational and NoSQL structures from a complex conceptual data model, which allowed to choose the type of data structures at the logical level and thus reduce database development time taking into account multivariate persistence.

Taking into account the improved structure of the conceptual data model and methods of automation of development of logical models of databases allowed to receive the fourth point of scientific novelty

- Improved conceptual modeling language AGILA MOD + due to the improved structure of the conceptual data model and methods of creating data models at the conceptual level, which allowed to take into account the multivariate persistence and reduce the design time of information system databases.

The *fourth chapter* develops an application programming interface (API) to provide unified access to databases with multivariate persistence in business logic.

The *fifth chapter* tests the conceptual modeling language AGILA MOD + and API in the development of databases in the process of IT projects and shows a

significant reduction in design time, which is confirmed by the relevant implementing acts.

Keywords: Database development, Polyglot Persistence, Modeling of Universe of Discourse (UoD), Conceptual Modeling language, Entity-Relationship, NoSQL database, Relational databases, Database Management System (DBMS)

СПИСОК ПУБЛІКАЦІЙ ЗДОБУВАЧА

1. Елена Арсирий, Matthias Kolonko, Sabine Müllenbach, Борис Трофимов, Мария Глава Concept modeling for developing databases with polyglot persistence as a direction of business digitalization. *Устойчивое развитие: Международный журнал*. (Варна, Болгария) – 2020. - №2-2020. – С. 22-29. Періодичний журнал Технічного університету м. Варна, Болгарія. <https://maurorg77.wixsite.com/maur-org/anotaciya>

2. Müllenbach Sabine, Kern-Bausch Lore, Kolonko Matthias. Conceptual Modeling Language Agila Mod. *Herald of Advanced Information Technology*, 2019 Vol. 2, No. 4, pp. 246-258. *Видання входить до міжнародних наукометричних баз Academia.edu, ROAD, Національна бібліотека України імені В. І. Вернадського, Djerelo, Україніка наукова, РИИЦ, Index Copernicus.* <https://hait.opu.ua/?fetch=articles&with=info&id=35>

3. Olena O. Arsirii., Maria G. Glava, Matthias Kolonko., Alina O. Glumenko. Information technology of supporting architectural solutions using polyglot persistence concept in learning management systems . *Applied Aspects of Information Technology*. 2020; Vol. 3 No. 2: – P 13-31 . *Видання входить до міжнародних наукометричних баз Academia.edu, ROAD, Національна бібліотека України імені В. І. Вернадського, Djerelo, Україніка наукова, РИИЦ, Index Copernicus* <https://aait.opu.ua/?fetch=articles&with=info&id=42>

4. M. Kolonko and S. Müllenbach, "Polyglot Persistence in Conceptual Modeling for Information Analysis," *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*, Deggendorf, Germany, 2020, pp. 590-594, doi:10.1109/ACIT49673.2020.9208928.

<https://ieeexplore.ieee.org/document/9208928> *Видання входить до міжнародних*

наукометричних баз **Scopus, IEEE Xplore**.

5. Matthias Kolonko and Sabine Müllenbach. Modern Databases – What’s really new about them?“ In: *Proceedings of the V. Ukrainian-German conference „Informatics. Culture. Technology.“ (IKT)*. (Odessa, Ukraine). Odessa National Polytechnical University, May 2017, pp. 35–37. ISBN: 978-966-2666-18-2

6. Matthias Kolonko and Sabine Müllenbach. „From E³R to AGILA MOD – A Syntax Evaluation and Advancement towards a Comprehensive Conceptual Semantically Irreducible Modeling Language.“ In: *Proceedings of Modern Information Technology 2018*. (Odessa, Ukraine). Odessa National Polytechnical University, May 2018, pp. 172–173. UDC: 004.55. ISBN: 978-617-7046-50-8

7. Matthias Kolonko and Sabine Müllenbach and Elena Arsirii and Boris Trofymov. „Extensions to the Conceptual Modeling Language AGILA MOD“. In: *Proceedings of the VI. Ukrainian-German conference „Informatics. Culture. Technology.“ (IKT)*. (Odessa, Ukraine). Odessa National Polytechnical University, September 2018, pp. 38–39. UDC: 004.652 ISSN: 2522-1523

8. Matthias Kolonko and Sabine Müllenbach. „An Analysis of Identification and Integrity Properties of Database Systems“ In: *Proceedings of the VII. Ukrainian-German conference „Informatics. Culture. Technology.“ (IKT)*. (Odessa, Ukraine). Odessa National Polytechnical University, September 2019, pp. 53–57. UDC: 004.652. ISSN: 2522-1523

9. Глуменко А.О., Стельмах Д.Е., Маттиас Колонко, Арсирій Е.А. Использование многовариантного хранения и мультимодельных СУБД для обработки разнообразных данных. Сучасні інформаційні технології : мат. X Міжнар. наук. конф. студентів та молодих вчених, м. Одеса, 14 – 15 травня 2020 р. / МОН України; Одес. Нац. політех. ун-т; Ін-т комп’ют. систем. Одеса : Наука і техніка, 2020. С. 118–119.