

УДК 004.94:681.883

DOI: 10.31673/2412-9070.2023.055665

Т. В. ПИРОГОВСЬКА, аспірантка;

А. М. ЄВТУШЕНКО,

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ

## СТРУКТУРА БАЗИ ЗНАНЬ МОДЕЛЮВАННЯ ГІДРОАКУСТИЧНИХ ПОЛІВ НА ОСНОВІ ПРОМЕНЕВОЇ ТЕОРІЇ

**Запропоновано дослідження щодо вдосконалення проєктування та документування реалізації моделювання за допомогою використання онтології. Модель онтології дає змогу структурувати інформацію, а також підвищувати ефективність пошуку. Розглянуто розроблення онтологічної моделі інженерних знань для моделювання гідроакустичних полів морської акваторії. Використання моделі онтології для розроблення порталу інженерних знань дає можливість систематизувати дані та знання, організувати пошук та навігацію, описувати інформаційні та обчислювальні ресурси відповідно до стандартів метапонять. Опис предметної галузі системи моделювання базується на онтології, що дає змогу оцінити повноту запропонованої моделі та забезпечити розширену детальну документацію системи.**

**Ключові слова:** Protege; онтологія; база знань; моделювання; гідроакустика; променева теорія; гідроакустичні поля акваторії.

### Вступ

**Постановка проблеми.** Програмування та інженерія знань мають багато спільного. Тоді як інженерія програмування намагається перейти до більшої абстракції у створенні програмного моделювання, інженерія знань розвинула кілька методів моделювання для реалізації семантичної мережі [1]. Зі стрімким розвитком вебзастосунків та вебсервісів задачі обох напрямків стали ще більш схожими. Дискусії щодо поєднання підходів у програмному забезпеченні та інженерії знань зазвичай мають наукове спрямування, зосереджуючись на окремих аспектах, зокрема метамоделюванні, і рідко надають корисних вказівок для розробників програмного забезпечення. Крім того, обидві сторони мають свої власні уявлення про ключові концепції, що може ускладнювати спілкування між їхніми учасниками.

Великі обсяги експериментальних даних, тривалий процес їх оброблення, корекції та статистичного опрацювання, на думку низки дослідників, об'єктивно зумовлюють потребу у створенні баз даних результатів досліджень. На сучасному рівні розвитку інформаційних технологій стає можливим формування не тільки баз даних, а й засобів інтеграції розподілених даних, результатів досліджень, завдань, сценаріїв, алгоритмів, методів оброблення погано структурованої інформації, математичних моделей тощо, що уможливають користувачам доступ до певних даних Access і різних ресурсів. Ці завдання потребують переходу на найсучасніше обладнання та використання інтернет-технологій. Відмінності у фізичних принципах і методах вимірювань, що лежать в основі дистанційних акустичних методів і стандартних гідрологічних спостережень, призводять до труднощів порівняно зі здобутими результатами. Адекватне моделювання складних процесів у географічному середовищі «вимагає інтенсивного використання знань із різних галузей науки і практики». Приклади окремих досліджень демонструють потребу в розробленні методів автоматизації ухвалення рішень щодо різноманітних задач гідроакустики на основі зібраних даних і знань. Слабка формалізація параметрів каналів зв'язку, нелінійність об'єктів керування, велика кількість непередбачуваних ситуацій, неможливість точного математичного опису інформації про властивості і характеристики об'єктів системного комплексу та середовища їх функціонування вимагають створення інтелектуальних систем керування, орієнтованих на концентрацію на поглинанні, обробленні та використанні інформації, що розуміються як знання [2].

**Аналіз останніх досліджень і публікацій.** Головною особливістю семантичної технології є збереження та підтримання цілісності семантики (сенсу знання), відокремленої від вмісту файлів даних і від коду програм, які їх реалізують. Технологія семантичного моделювання відрізняється від традиційних методів тим, що вона поєднує значення даних і процедури оброблення безпосередньо в програмному коді. Це часто призводить до потреби в радикальному ручному реінжинірингу структур даних і повному перегляду програм під час їх розроблення або міграції на іншу платформу. Приклади використання онтологічної моделі в галузі технологій Семантичні технології дають змогу отримувати логічні висновки на основі правил концептуальних моделей і здійснюють автоматичне перепроєктування структури даних.

Досвід семантичного моделювання інтелектуальних систем за допомогою онтологій показує, що будь-яка предметна сфера може бути описана значною кількістю онтологій. З методологічного погляду

© Т. В. Пироговська, А. М. Євтушенко, 2023

це цілком зрозуміло — кожна онтологія відображає уявлення людини про розробника моделі функціонування онтології (основні одиниці, класи, підкласи та їх зв'язки в межах загального уявлення про предметну сферу). Тому розробнику доцільно використовувати таку методологію та інструменти, які дають змогу не тільки розробити модель онтології, а й коригувати її в процесі розроблення та розуміння особливостей її функціонування, спрямованих на коректне розроблення моделі, наскільки це можливо [3].

Яку роль можуть відігравати онтології в галузі розроблення програмного забезпечення? Вольфганг Гессе у своїй праці [4] виокремлює наведені далі два тісні зв'язки.

• *Онтології процесу розроблення програмного забезпечення*: модуляризація, поширення, повторне використання та інтеграція програмних компонентів і систем є одним із центральних питань програмної інженерії з перших днів. Чим більше ці завдання розширюються та автоматизуються, тим важливішим стає визначення та використання онтологій як концептуальної основи таких компонентів. Сучасні методології розроблення програмного забезпечення, зокрема розроблення, кероване моделями (MDD), віддають перевагу моделям як центральній базі знань, з якої можуть бути отримані різні реалізації. Якщо системи або компоненти мають обмінюватися знаннями, це відбуватиметься більше на рівні моделювання, ніж на рівні впровадження. В епоху об'єднання ландшафтів застосунків розвивається великий потенціал для визначення, впровадження, поширення та використання онтологій.

• *Онтології для галузі програмної інженерії*: програмна інженерія є власною науковою та професійною галуззю з власною структурою та термінологією. Оскільки це досить нова сфера, завдання формування однієї (чи кількох) онтологій для цієї галузі є необхідним і складним завданням. Можна згадати два підходи: проєкт SWEBoK в англійській спільноті і німецьку термінологічну мережу («Begriffsnetz»).

У своїй статті [5] науковці вперше описали створення DEMO-процесора за допомогою онтологічного моделювання. DEMO-процесор у повному обсязі — це адаптивна система керування справами для комунального підприємства. Виробництво охоплює складні документальні контракти, підпорядковані правилам бізнесу. Цей випадок показує можливість реалізації як виробничої IT-системи. Проблема узгодження бізнесу та IT вирішується за допомогою онтології з підтвердженням відповідності. Пряме виконання онтологічних моделей забезпечує для цього хороший ступінь узгодженості бізнес-IT EIS. Можливість піддавати моделі перевірці на дуже ранній стадії проєкту також підтримує високий рівень узгодженості бізнесу та IT. Проблема неконтрольованих витрат на програмування програмного забезпечення розв'язується прямим виконанням моделей DEMO як здійснюваного вихідного коду процесора DEMO. У першій професійній програмі процесора DEMO програмування програмного забезпечення обмежується інтерфейсами із застарілими системами тощо; основна програма створюється з моделей. Погіршення якості програмного забезпечення під час поточних модифікацій було усунуто, оскільки DEMO-процесор Enterprise Ontology Driven Software Engineering 209 строго відповідає вимогам нормалізованих систем (Mannaert, Verelst, 2008). Процесор DEMO також пропонує повні можливості робочого процесу, усунення аномалій, зокрема взаємоблокування (Nuffel, 2009), а отже, будь-яке BPM (подібне) моделювання є застарілим.

Також у своїй праці [6] Мюнсен описує використання Protégé для створення комп'ютерної системи запису пацієнтів, яка допомагає постачальникам у застосуванні рекомендацій і протоколів клінічної практики. Показано, як підхід вирішує принципову побудову онтології предметної галузі, інстанціювання цієї онтології з фактами домену та створення прикладної програми, яку розробники можуть легко підтримувати та змінювати з часом. Автор наголошує на зв'язку між стандартними, контрольованими термінологіями та онтологіями домену, які формують основу побудови прикладних систем за допомогою Protégé. У цій статті продемонстровано використання явних онтологій домену у формуванні всіх аспектів процесу програмного забезпечення та описано, як залучення цих онтологій безпосередньо до програмних систем може зумовити вдосконалене розроблення та покращене обслуговування системи [6].

Важливим кроком у процесі розроблення є перехід від аналізу та проєктування до впровадження. Для цього ключову роль завжди відігравав спосіб, в який проблемна ділянка відображалася в коді. Постає питання, як можна використати онтології, щоб скоротити розрив між дизайном і реалізацією. Сферами інтересу є перекриття моделювання програмного забезпечення з мовами онтології та застосування онтологій у програмах під час виконання, а також методи, де онтології підтримують кодування та документацію коду.

Оскільки моделювання онтологій є виснажливим і дорогим завданням, завжди важливо продемонструвати переваги, які можна дістати, застосовуючи онтології в розробленні програмного забезпечення. Це підкреслюється тим фактом, що більшість формальних основ онтологій існують протягом

тривалого часу, не набувши широкого застосування розробниками програмного забезпечення. Отже, очевидно, що нинішня поява формалізмів на основі логіки в контексті зусиль семантичної мережі є важливим фактором. Діяльність W3C та інших допомогла конкретизувати такі стандарти, як RDF або OWL, котрі привертають увагу розробників інструментів і користувачів. У певному сенсі важливість стандартизації тут можна порівняти з ситуацією візуального моделювання в розробленні програмного забезпечення до UML.

Іншим вагомим фактором є гнучкість онтологій. Оскільки інтеграція інформації є основним випадком використання, онтології добре підходять для поєднання інформації з різних джерел і отримання нових фактів на її основі. Крім того, гнучкість дає змогу дуже легко розширювати теперішні онтології, у такий спосіб сприяючи повторному використанню наявної роботи. Цьому також сприяє «вебфокус» сучасних онтологічних підходів. У зв'язку з тим, що системи програмного забезпечення також стають дедалі активнішими для роботи з інтернетом і тому мають працювати з даними з різномірних джерел, які можуть бути невідомі під час розроблення, розробники програмного забезпечення шукають технології, які можуть допомогти в цій ситуації. Отже, такі експерти в цій галузі, як Грейді Буч, очікують, що семантична вебтехнологія стане однією з подальших великих речей в архітектурі вебзастосунків [7]. Крім того, інтернет полегшує обмін знаннями. Маючи URI як глобальні унікальні ідентифікатори, можна легко пов'язати свою онтологію з чиеюсь концептуалізацією. Це, зі свого боку, сприятиме взаємодії та повторному використанню. Що стосується більш специфічних переваг програмної інженерії, то онтології роблять моделі домену громадянами першого порядку. Незважаючи на те, що моделі предметної сфери явно є основою кожної програмної системи, їхня важливість у поточних процесах розроблення програмного забезпечення зменшується після фази аналізу. Основною метою онтологій за визначенням є формальний опис домену, що сприяє більш широкому використанню протягом усього життєвого циклу розроблення програмного забезпечення [8].

**Метою статті** є розроблення та дослідження підходу до побудови онтологічної бази знань для опису, аналізу, формалізації неоднорідних гідроакустичних інформаційних ресурсів та побудови сценаріїв дослідження для створення документаційної бази програмного забезпечення.

### Основна частина

Онтологічний метод є зручним способом детальної документації програмного продукту, яка охоплює не тільки роз'яснення архітектури продукту, а також дає змогу закріплювати теоретичну складову за кожною сутністю. Такий спосіб документації полегшує ведення нових працівників у проєкт, а також уможливорює перегляд поточної структури проєкту кожної ітерації розроблення програмного забезпечення.

Розглянемо документацію проєкту реалізації моделювання гідроакустичних полів морської акваторії на основі променевої теорії. Діаграму класів розробленого програмного забезпечення, реалізованого мовою C#, у вигляді динамічно приєднуваної бібліотеки класів зображено на рис. 1.

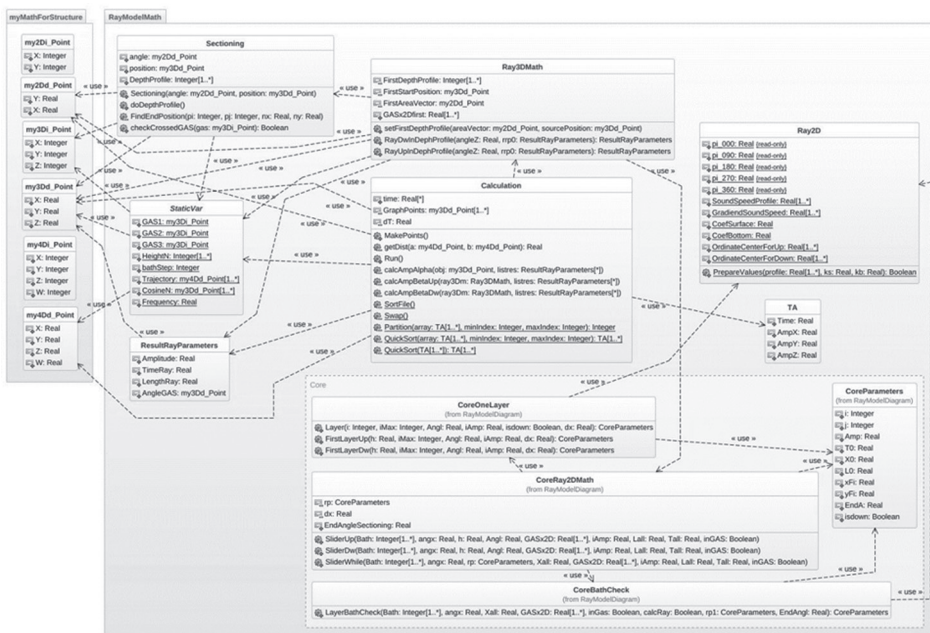


Рис. 1. Діаграма класів програмного забезпечення моделювання гідроакустичних полів морської акваторії на основі променевої теорії

Бібліотека класів RayModelMath.dll складається з 11 класів і також використовує додаткові шість структур, в яких реалізовано 2D-, 3D- та 4D-вектори (див. рис. 1). У головній програмі задаються значення властивостям класу StaticVar, які є вхідними даними для розрахунку: частота сигналу, координати розміщення ГАС, масив глибин у вузлових точках та крок сітки батиметрії водного середовища, масив точок траєкторії руху об'єкта. Головний клас, з якого починається розрахунок, має назву Calculation. У класі Ray3DMath реалізовано проміжні дії, що відбуваються між запуском розрахунку та безпосередньо самим розрахунком. У класі Sectioning реалізовано розрахунок вузлових точок перерізу батиметрії водного середовища. Основний розрахунок траєкторії звукового променя реалізовано в класах CoreRay2DMath, CoreOneLayer, CoreBathCheck.

Клас CoreBathCheck містить лише один метод LayerBathCheck(int[], double, double, double[], ref bool, ref bool, CoreParameters, ref double). Метод має вісім параметрів: масив глибин у вузлових точках батиметрії, крок перерізу батиметрії, проекцію всієї траєкторії променя на горизонтальну вісь, масив відстаней від початкової точки перерізу до ГАС, два параметри, що зазначають, чи влучив промінь у ГАС та чи потрібно продовжувати розрахунок у цьому перерізі батиметрії, об'єкт класу CoreParameters, в якому зберігаються результати обчислень одного з методів (FirstLayerUp(), FirstLayerDw() або Layer()), та останній параметр, в який записується обчислений кут потрапляння променя в ГАС. У методі LayerBathCheck() перевіряється, чи відбулось влучання променя в похиле дно або в ГАС, внаслідок якого розрахунок траєкторії променя зупиняється. Для ініціалізації статичних властивостей у класі Ray2D створено метод PrepareValues(double[], double, double), який приймає три параметри: масив значень профілю швидкості звуку, коефіцієнти загасання амплітуди під час відбиття звукового променя від поверхні та від дна водного середовища. У методі відбувається розрахунок градієнтів швидкості звуку та координати Y центрів кіл для кожного водного прошарку. Глобальні параметри розрахунку, які використовуються методами одразу кількох класів, реалізовано як властивості статичного класу StaticVar.

C# — об'єктно-орієнтована мова, тому виокремлення сутностей, об'єктів, параметрів і дата типів значно спрощується. Визначення класів і створення їх ієрархії (таксономії) є ключовими в розробленні онтології. Таксономія класів — це дерево описових термінів, які мають ієрархічну структуру. У редакторі Protégé 5 створення класів відбувається у закладці «Classes». В OWL класи інтерпретуються як підмножина індивідів, котрі є частиною визначеного класу. Особливістю проектування в середовищі Protégé є те, що класи розглядаються як підкласи загальної онтології THING. Відповідно до анотації CamelCase для OWL [12] усі імена класів мають починатися з великої літери і не повинні містити пропусків. За замовчуванням класи в OWL можуть перетинатися. Щоб розділити класи, їх потрібно роз'єднати. Це гарантує, що окрема особа не може бути екземпляром більш ніж одного класу. OWL дає змогу розробляти анотації з різною інформацією (коментарями, датами створення, автором, посиланнями на ресурси тощо) і класами метаданих, властивостями, особами та онтологіями. Це дає можливість детально описувати кожен створений клас, його властивості та взаємодії з іншими класами системи. Ієрархію та зв'язки створених сутностей зображено на рис. 2.

Підмножина атрибутів описує властивості класів і використовується для введення конкретних значень екземплярів у класи. Створення підмножини атрибутів редактором Protégé було виконано за допомогою вкладки на панелі інструментів редактора Protégé 5. Далі відображається вікно, що містить вкладку. Оскільки вибрана зона достатньо складна, для зручності всі атрибути було поділено як ієрархічну структуру (рис. 3). Такий спосіб групування атрибутів спрощує читання властивостей, а також прискорює пошук потрібних у базі знань. Попередньо визначені атрибути зазначено в словнику схеми XML і можуть бути подані в різних форматах даних, таких як цілі числа, з рухомою комою, рядки, логічні значення тощо.

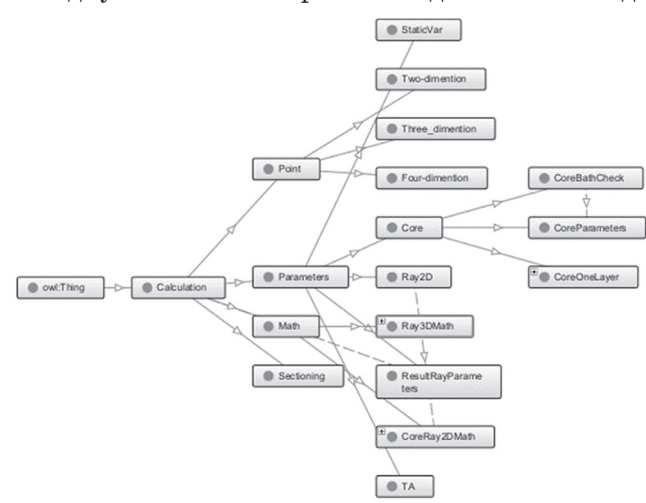


Рис. 2. Створена онтологія реалізації моделювання гідроакустичних полів акваторії

У редакторі Protégé кожен клас визначається властивостями, які окреслюють зв'язок між класами, а властивості об'єкта, що описують вкладкою «ObjectProperty» (рис. 4), є відносинами між двома класами та окремими особами.

За допомогою вкладки <ObjectProperty> можна схарактеризувати такі зв'язки:

- формування підмножини зв'язку між класами у визначеній предметній сфері через додавання типів зв'язків до списку;

- формування підмножини аксіом у визначеній предметній сфері завдяки зв'язуванню певних зв'язків онтології класу за допомогою предикатів із «owl: topObjectProperty»;

- формування відносин між особами за допомогою предикатів, що містяться в «owl: topObjectProperty».

Основні взаємодії класів, які не є прямими під-класами або класами-батьками, унаочнює рис. 4.

### Висновки

Побудована семантична модель дала можливість детально проаналізувати визначену предметну сферу та якість реалізації архітектури програмного забезпечення. Створення та оновлення онтології в процесі розроблення дає змогу контролювати поточний стан системи та уникати непотрібного її ускладнення. Тому онтологічний підхід до побудови архітектури програмного забезпечення є гарною альтернативою простому опису моделі за допомогою UML-діаграм, оскільки зберігає інформацію про всю систему в одній базі знань, яка здатна продукувати різноманітні візуалізації залежностей усіх сутностей моделі.

У статті детально описано створення бази знань на основі проведеного моделювання гідроакустичних полів морської акваторії. Базу знань призначено для комплексного інформаційного забезпечення теоретичних та експериментальних досліджень у галузі гідроакустики, основним завданням яких є автоматизоване отримання, інтеграція, формалізація даних і знань в інтересах як підготовки фахівців у навчальному процесі, так і проведення фундаментальних наукових досліджень.

### Список використаної літератури

1. **Berners-Lee T., Hendler J., Lassila O.** *The Semantic Web* // *Scientific American* 2001. 284(5).
2. **Scientific research ontology to support systematic review in software engineering** / J. Calmon de Almeida Biolchini, P. G. Mian, Ana Candida Cruz Natali [et al.] // *Advanced Engineering Informatics*. 2007. Vol. 21, Issue 2. P. 133–151.
3. **Examples of Ontology Model Usage in Engineering Fields** / L. Globa, R. Novograduska, A. Koval, V. Senchenko // *Ontology in Information Science*. 2018. InTech, Mar. 08.
4. **Hesse W.** *Ontologies in the Software Engineering Process*. 2005.
5. **Enterprise ontology driven software engineering** / Steven JH. van Kervel [et al.] // *International Conference on Software Paradigm Trends*. 2012. Vol. 2. SCITEPRESS.
6. **Musen M. A.** *Domain ontologies in software engineering: use of Protégé with the EON architecture* // *Methods Inf Med*. 1998. P. 37(4-5):540–50.
7. **Booch G.** *Generations of Software Architecture*. 2005.
8. **Happel H. J., Seedorf S.** *Applications of ontologies in software engineering* // *Proc. of Workshop on Semantic Web Enabled Software Engineering (SWESE) on the ISWC*. 2006. P. 5–9.
9. **Jureta I., Mylopoulos J., Faulkner S.** *Revisiting the Core Ontology and Problem in Requirements Engineering* // *RE*. 2008. P. 71–8.
10. **Koval S.** *Automatic text processing based on the object predicate system* // *Structural and applied linguistics*. 1998. P. 199–207.
11. **METHONTOLOGY: From Ontological Art Towards Ontological Engineering** / M. Ferndndez, A. Gomez-Perez, N. Juristo. URL: [http://oa.upm.es/5484/1/METHONTOLOGY\\_.pdf](http://oa.upm.es/5484/1/METHONTOLOGY_.pdf)
12. **Semantic Web Technologies**. 2013 [Електроний ресурс]. URL: <https://open.hpi.de/course/semanticweb>

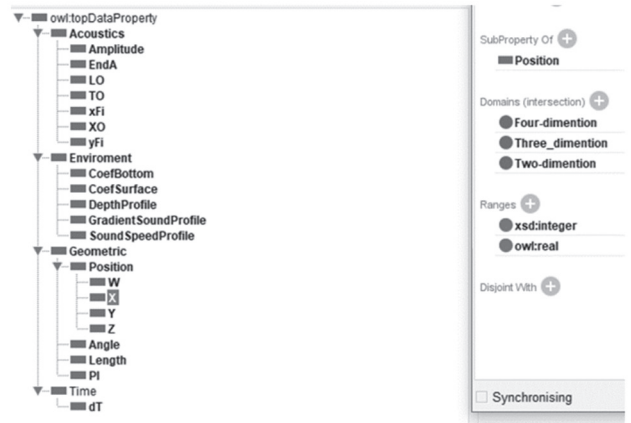


Рис. 3. Властивості об'єктів системи

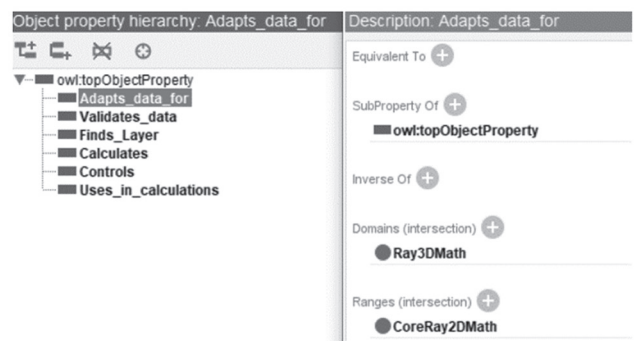


Рис. 4. Опис взаємодії класів системи

13. *OWL 2 Web Ontology Language Document Overview: Second Edition. W3C Recommendation. 11 December 2012.* [Електронний ресурс]. URL:  
<http://www.w3.org/TR/owl2-over>

*T. Pyrohovska, A. Yevtushenko*

### **STRUCTURE OF THE KNOWLEDGE BASE FOR MODELING HYDROACOUSTIC FIELDS BASED ON RAY THEORY**

*The proposed research suggests improving the design and documentation of the implementation of modeling through the use of ontology. The ontology model allows you to structure information and also increases search efficiency. The study describes the development of an ontological model of engineering knowledge for modeling the hydroacoustic fields of the marine water area. The use of the ontology model for the development of the engineering knowledge portal allows you to systematize data and knowledge, organize search and navigation, and describe information and computing resources in accordance with metaconcept standards. The description of the subject area of the modeling system is based on the ontology, which allows you to assess the completeness of the proposed model and provide extended detailed documentation of the system.*

**Keywords:** Protege; ontology; knowledge base; modeling; hydroacoustics; ray theory; hydroacoustic fields of the water area.

---

## **ЗВ'ЯЗОК**

*Наукове фахове видання*

Редакційна обробка та коректура

*Т. В. Ількевич*

Комп'ютерна верстка та дизайн

*Г. С. Тимченко*

Відповідальний за випуск

*І. І. Тищенко*

Формат 60×84/8. Папір друкарський.  
Гарнітура SchoolBookC, EuropeCond. Зам. 900  
Наклад 300 прим.

Державний університет інформаційно-комунікаційних технологій  
03110, м. Київ, вул. Солом'янська, 7  
Тел. (044) 249-25-75  
E-mail: [zviaz-ok@ukr.net](mailto:zviaz-ok@ukr.net)