

УДК 004.891

О. В. Бісікало, О. В. Кудрик

БАЗА ЗНАТЬ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ФАЗОВОЇ СТАБІЛЬНОСТІ ТВЕРДИХ РОЗЧИНІВ

Вінницький національний технічний університет, Вінниця

Анотація. Дана дослідницька робота присвячена розробці бази знань для розв'язання актуальної задачі прогнозування фазової стабільності твердих розчинів. Під базами знань розуміється сукупність фактів і правил виводу, що допускають логічний висновок і цілеспрямовану обробку інформації. Найбільш важливою властивістю інформації, що зберігається в базах знань, є достовірність конкретних і узагальнених відомостей у базі даних і релевантність вихідної інформації, одержуваної з використанням правил виведення, закладених в базу знань. Кращі бази знань включають саму релевантну, достовірну і свіжу інформацію, мають досконалі системи пошуку інформації, ретельно продуману структуру і формат знань. Експертна система втілює методологію адаптації алгоритму успішних рішень однієї сфери науково-практичної діяльності в іншу. З поширенням комп'ютерних технологій – це тотожна інтелектуальна комп'ютерна програма, що містить знання й аналітичні здібності одного чи кількох експертів в деякій галузі застосування і здатна робити логічні висновки на основі цих знань, тим самим забезпечуючи вирішення специфічних завдань. Інтелектуальна інформаційна система (ІС) – це один з видів автоматизованих інформаційних систем, який являє собою комплекс програмних, лінгвістичних і логіко-математичних засобів для реалізації основного завдання, яке зазвичай полягає у інтерпретації даних та прогнозуванні. Інтерпретація даних – це одне з традиційних завдань для експертних систем. Під інтерпретацією розуміється процес визначення змісту даних, результати якого мають бути погодженими і коректними. Зазвичай передбачається багатоваріантний аналіз даних, а виведені з цієї моделі сліdstва складають основу для прогнозів з ймовірними оцінками. Прогнозування дозволяє передбачати наслідки деяких подій або явищ на підставі аналізу наявних даних. У прогнозуючій системі зазвичай використовується параметрична динамічна модель, в якій значення параметрів встановлюються під задану ситуацію. У результаті розробки було побудовано модель бази знань для прогнозування фазової стабільності твердих розчинів за допомогою множини продукційних правил, предикатів, функцій та операторів.

Ключові слова: інтелектуальна інформаційна система, база знань, продукційні правила, предикати, оператори.

Abstract. This research work is devoted to the development of a knowledge base for solving the current problem of forecasting the phase stability of solid solutions. Knowledge bases mean a set of facts and inference rules that allow logical conclusion and purposeful processing of information. The most important property of information stored in knowledge bases is the reliability of specific and generalized information in the database and the relevance of the original information obtained using the rules of inference embedded in the knowledge base. The best knowledge bases include the most relevant, reliable and fresh information, have perfect information search systems, a carefully thought-out structure and format of knowledge. The expert system embodies the methodology of adapting the algorithm of successful solutions from one sphere of scientific and practical activity to another. With the spread of computer technologies, it is an identical intelligent computer program that contains the knowledge and analytical abilities of one or more experts in some field of application and is able to draw logical conclusions based on this knowledge, thereby providing a solution to specific tasks. An intelligent information system (IIS) is one of the types of automated information systems, which is a complex of software, linguistic and logical-mathematical tools for the implementation of the main task, which usually consists of data interpretation and forecasting. Data interpretation is one of the traditional tasks for expert systems. Interpretation means the process of determining the content of data, the results of which must be agreed and correct. A multivariate analysis of the data is usually assumed, and the findings from this model form the basis for probabilistic estimates. Forecasting allows you to predict the consequences of some events or phenomena based on the analysis of available data. A parametric dynamic model is usually used in the forecasting system, in which parameter values are set for a given situation. As a result of the development, a knowledge base model was built for predicting the phase stability of solid solutions using a set of production rules, predicates, functions and operators.

Key words: intellectual information system, knowledge base, production rules, predicates, operators.

DOI: <https://doi.org/10.31649/1999-9941-2023-56-1-13-21>.

Вступ

База знань – сукупність систематизованих основних відомостей, що стосуються певної галузі знань і засобів, за допомогою яких відбуваються накопичення, збереження, оновлення та використання знань.

Відмінністю бази знань від бази даних є можливість формування нових знань [1].

Головні функції бази знань стосовно систем автоматизованого проектування (САПР):

- опис предметної галузі САПР;
- підтримка інтелектуальних методів розв'язання задач, що входять до складу САПР;
- реалізація можливостей експертного аналізу проектних завдань.

Продукційні правила – це форма представлення знань людини у вигляді речення типу ЯКЩО (умова), ТО (дія). Правила забезпечують формальний спосіб представлення рекомендацій, вказівок або стратегій. Вони ідеально підходять в тих випадках, коли знання предметної області виникають з емпіричних асоціацій, накопичених за роки роботи по вирішенню завдань у тій чи іншій галузі.

Продукційна модель – це набір продукційних правил, яка, з одного боку, близька до логічних моделей, що дозволяє організувати на ній ефективні процедури виведення, а з іншого боку, більш наочно відображає знання.

Продукційні правила використовуються в системах штучного інтелекту (як приклад, експертні системи) [2], як одна з найпоширеніших форм представлення знань, на ряду з логічними моделями, фреймами та семантичними мережами [3].

Експертна система – це методологія адаптації алгоритму успішних рішень однієї сфери науково-практичної діяльності в іншу. З поширенням комп'ютерних технологій – це тотожна (подібна, заснована на оптимізуючому алгоритмі чи евристичних) інтелектуальна комп'ютерна програма, що містить знання й аналітичні здібності одного чи кількох експертів в деякій галузі застосування і здатна робити логічні висновки на основі цих знань, тим самим забезпечуючи вирішення специфічних завдань (консультування, навчання, діагностування, тестування, проектування тощо) без участі експерта (фахівця в конкретній проблемній галузі). Визначається також як система, яка використовує базу знань для вирішення завдань (видачі рекомендацій) у певній предметній галузі. Цей клас програмного забезпечення спочатку розроблявся дослідниками штучного інтелекту в 1960-ті та 1970-ті та здобув комерційне застосування, починаючи з 1980-х. Часто термін система, заснована на знаннях, використовується як синонім експертної системи, однак можливості експертних систем ширші за можливості систем, заснованих на детермінованих (обмежених, реалізованих на поточний час) знаннях [4].

Інтелектуальна інформаційна система (ІІС) – це один з видів автоматизованих інформаційних систем, інколи ІІС називають системою, засновану на знаннях. ІІС є комплексом програмних, лінгвістичних і логіко-математичних засобів для реалізації основного завдання: здійснення підтримки діяльності людини і пошуку інформації в режимі розширеного діалогу природною мовою [5].

Мета

Мета статті – розробка бази знань для інтелектуальної інформаційної системи прогнозування фазової стабільності твердих розчинів.

Побудова бази знань

Розглянемо модель бази знань продукційного типу для прогнозування фазової стабільності твердих розчинів, що складається з множини відношень, продукційних правил, предикатів, функцій та операторів. Представимо модель, що розробляється, як

$$KnowledgeBase = \langle Re, Rule, Pr, Func, Op \rangle ,$$

що складається з відношень RE , продукційних правил

$$Rule = \{ ParamsR, ParamsChargeR, ParamsChargeCoordinationR, TypeAddR, TypeAdd \},$$

предикатів $Pr = \{ A(p_i), B(p_{c_j}), C(p_{cc_n}), TypeAdd_j, Type_j, PA(a_j), AD(s_d), AL(prm_j) \},$

функцій

$$Func = \{ ElementParams, ElementParamsCharge, ElementParamsCoordination, ElementTypeAdd, ElementType, PredAnalysis, AnalysisData, PredTypeAlgorithm \},$$

та операторів

$$Op = \{ Oprint, Oanalysis, OdataCheck, Opred, OsaveResult, OreturnResult \}.$$

Формальними відношеннями, які дозволяють розглядати інформаційну складову «Інтелектуальної системи фазової стабільності твердих розчинів» у термінах реляційної моделі даних будемо вважати:

$$RE = \{ element_grp, sub_element_grp, element, elm2elm, charge_element, cordination_element, atom_length, sum_atom_length, volume_cell, structure_solid, term_system, tsys_crde, stored_system, stored_result \}, \quad (1)$$

де, $element_grp$ – відношення для характеристики групи елементів; атрибути, з яких складається це відношення, позначають: elm_id (унікальний ід-код певної групи елементів), elm_name (назва групи елементів).

$$Element_grp \subset elm_id \times elm_name \quad (2)$$

$sub_element_grp$ – відношення для характеристики підгрупи елементів; атрибути, з яких складається це відношення, позначають: $selm_id$ (унікальний ід-код певної підгрупи елементів), elm_name (назва підгрупи елементів).

$$Sub_element_grp \subset selmg_id \times selmg_name \quad (3)$$

Element – відношення для характеристики підгрупи елементів; атрибути, з яких складається це відношення, позначають: *elm_id* (унікальний ід-код певного елемента), *elm_selmg* (ідентифікатор підгрупи), *elm_name* (назва елемента), *elm_code* (хімічний код), *elm_count* (кількість атомів елемента).

$$Element \subset elm_id \times elm_selmg \times elm_name \times elm_code \times elm_count \quad (4)$$

Elm2elm – відношення для характеристики складного елемента; атрибути, з яких складається це відношення, позначають: *e2e_id* (унікальний ід-код певного складного елемента), *e2e_che_par* (ідентифікатор заряду батьківський), *e2e_che_ch* (ідентифікатор заряду дочірній), *e2e_sort* (сортування).

$$Elm2elm \subset e2e_id \times e2e_che_par \times e2e_che_ch \times e2e_sort \quad (5)$$

Charge_element – відношення для характеристики зарядів елементів; атрибути, з яких складається це відношення, позначають: *che_id* (унікальний ід-код певного заряду елемента), *che_elm* (ідентифікатор елемента), *che_value* (значення заряду).

$$Charge_element \subset che_id \times che_elm \times che_value \quad (6)$$

Cordination_element – відношення для характеристики координаційних номерів зарядів; атрибути, з яких складається це відношення, позначають: *crde_id* (унікальний ід-код певного координаційного номера заряду), *crde_che* (ідентифікатор заряду елемента), *crde_value* (значення координаційного числа).

$$Cordination_element \subset crde_id \times crde_che \times crde_value \quad (7)$$

Atom_length – відношення для характеристики атомних довжин між зарядами елементів; атрибути, з яких складається це відношення, позначають: *atml_id* (унікальний ід-код певної атомної довжини між зарядами елементів), *atml_che_from* (ідентифікатор заряду елемента 1), *atml_che_to* (ідентифікатор заряду елемента 2) *atml_value* (значення атомних довжин), *atml_strs* (ідентифікатор структури твердого розчину).

$$Atom_length \subset atml_id \times atml_che_from \times atml_che_to \times atml_value \times atml_strs \quad (8)$$

Sum_atom_length – відношення для характеристики суми атомних довжин між зарядами елементів; атрибути, з яких складається це відношення, позначають: *satml_id* (унікальний ід-код певної суми атомної довжини між зарядами елементів), *satml_che_from* (ідентифікатор заряду елемента 1), *satml_che_to* (ідентифікатор заряду елемента 2), *satml_value* (значення сум атомних довжин), *satml_strs* (ідентифікатор структури твердого розчину).

$$Sum_atom_length \subset satml_id \times satml_che_from \times satml_che_to \times satml_value \times satml_strs \quad (9)$$

Volume_cell – відношення для характеристики обсягу комірок між зарядами елементів; атрибути, з яких складається це відношення, позначають: *volc_id* (унікальний ід-код певного обсягу комірок між зарядами елементів), *volc_che_from* (ідентифікатор заряду елемента 1), *volc_che_to* (ідентифікатор заряду елемента 2), *volc_value* (значення обсягу) *volc_strs* (ідентифікатор структури твердого розчину).

$$Volume_cell \subset volc_id \times volc_che_from \times volc_che_to \times volc_value \times volc_strs \quad (10)$$

Structure_solid – відношення для характеристики структури твердого розчину; атрибути, з яких складається це відношення, позначають: *strs_id* (унікальний ід-код певної структури розчину), *strs_name* (назва групи елементів).

$$Structure_solid \subset strs_id \times strs_name \quad (11)$$

Term_system – відношення для характеристики термодинамічної системи; атрибути, з яких складається це відношення, позначають: *tsys_id* (унікальний ід-код певної термодинамічної системи), *tsys_crde* (ідентифікатор координаційного елемента), *tsys_elm* (ідентифікатор елемента).

$$Term_system \subset tsys_id \times tsys_crde \times tsys_elm \quad (12)$$

Tsys_crde – відношення для характеристики елементів, пов’язаних з певною термодинамічною системою; атрибути, з яких складається це відношення, позначають: *tsysc_id* (унікальний ід-код певного елемента термодинамічної системи), *tsysc_tsys* (ідентифікатор термодинамічної системи), *tsysc_crde* (ідентифікатор координаційного елемента).

$$Tsys_crde \subset tsysc_id \times tsysc_tsys \times tsysc_crde \quad (13)$$

Stored_system – відношення для характеристики дані системи, яка була розрахована; атрибути, з яких складається це відношення, позначають: *stds_id* (унікальний ід-код певних даних про систему, яка була розрахована), *stds_strs* (ідентифікатор структури системи), *stds_ln1* (ідентифікатор лантаноїда 1), *stds_ln2* (ідентифікатор лантаноїда 2), *stds_anion* (ідентифікатор аніона), *stds_eps* (крок обчислень).

$$Stored_system \subset stds_id \times stds_strs \times stds_ln1 \times stds_ln2 \times stds_anion \times stds_eps \quad (14)$$

Stored_result – відношення для характеристики дані про розрахунки системи; атрибути, з яких складається це відношення, позначають: *stdr_id* (унікальний ід-код певних даних про розраховану систему), *stdr_std* (ідентифікатор розрахованої системи), *stdr_x* (результат x), *stdr_x1* (результат x1), *stdr_x2* (результат x2), *stdr_t* (результат t), *stdr_t_crit* (результат t_crit), *stdr_q* (результат q), *stdr_method* (результат method), *stdr_err_message* (текст помилки).

$$Stored_result \subset stdr_id \times stdr_stds \times stdr_x \times stdr_x1 \times stdr_x2 \times stdr_t \times stdr_t_crit \times stdr_q \times stdr_method \times stdr_err_message \quad (15)$$

Схему формальних відношень можна побачити на рисунку 1 [6].

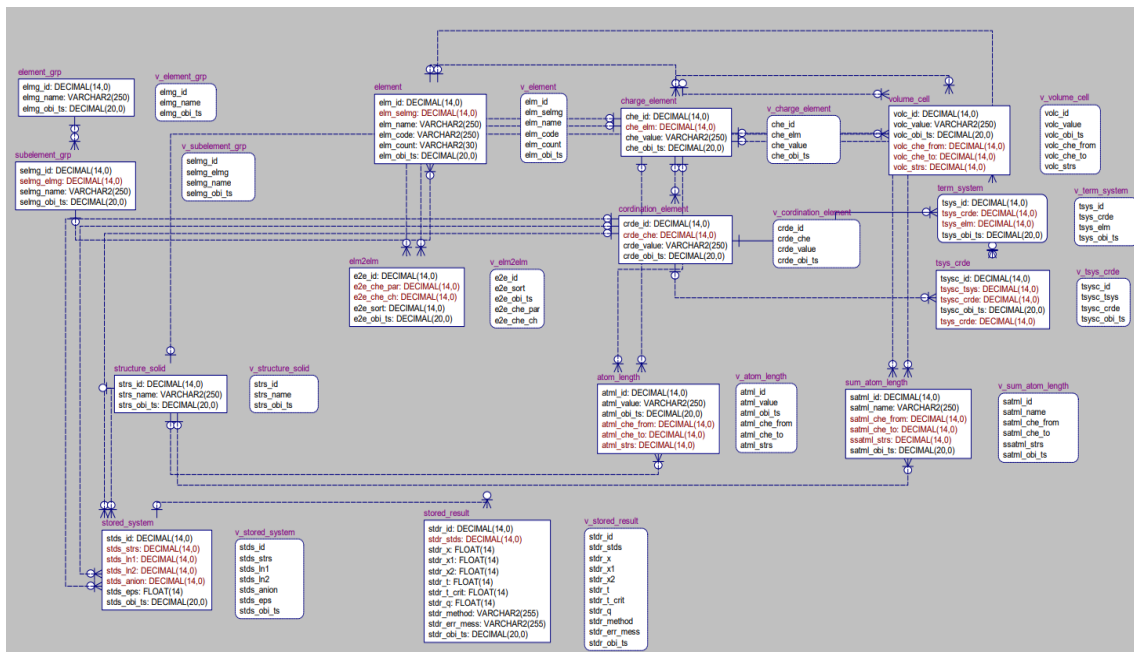


Рисунок 1 – Схема формальних відношень

Під терміном елемент будемо розуміти складову, яка визначається через певні параметри. У загальному випадку параметри змінюються в залежності від виду елемента. Використовуючи *id* елемента з таблиці *element*, можливо з’єднати декілька елементів, які у подальшому формують складний елемент. Для цього розроблено базу знань продукційного типу інтелектуального модуля “Елементи”, що складається з множини продукційних правил, предикатів, функцій та операторів. Зокрема:

1. В процесі створення елемента враховано можливість внесення загальних параметрів до елемента (функція *ElementParams*). Так, для кожного елемента потрібно забезпечити отримання показників з бази даних. Отже, якщо параметр r_i деякого i -го елемента з N відсутній в системі, обирається універсальне значення r_{def} , інакше необхідно ввести значення r . Формально визначимо продукцію *ParamsR*:

Якщо $A(p_i)$ тоді $ElementParams(p_i)$ інакше $ElementParams(p_{def})$, де предикат $A(p_i)$ визначається так:

$$\exists p_i \mid i \in \{1, 2, \dots, N\} \rightarrow A(p_i) = True$$

На рисунку 2 показано приклад введення загальних параметрів.

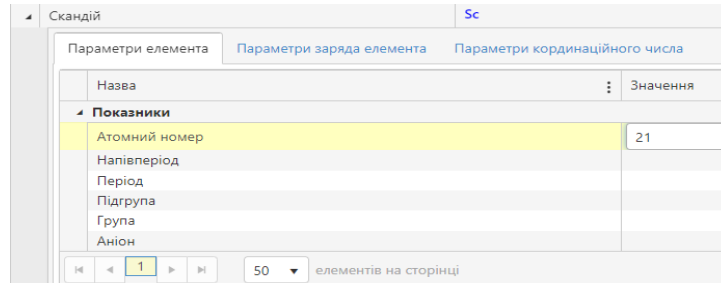


Рисунок 2 – Введення загальних параметрів

2. В процесі створення елемента враховано можливість внесення параметрів заряду до елемента (функція $ElementParamsCharge$). Так, для кожного елемента потрібно забезпечити отримання показників з бази даних. Отже, якщо параметр pc_j деякого j -го елемента з M відсутній в системі, обирається універсальне значення pc_{def} , інакше необхідно ввести значення pc_j . Формально визначимо продукцію $ParamsChargeR$:

Якщо $B(pc_j)$ тоді $ElementParamsCharge(pc_j)$ інакше $ElementParamsCharge(pc_{def})$, де предикат $B(pc_j)$ визначається так:

$$\exists pc_j \mid j \in \{1, 2, \dots, M\} \rightarrow B(pc_j) = True$$

На рисунку 3 показано приклад введення параметрів заряду елемента.

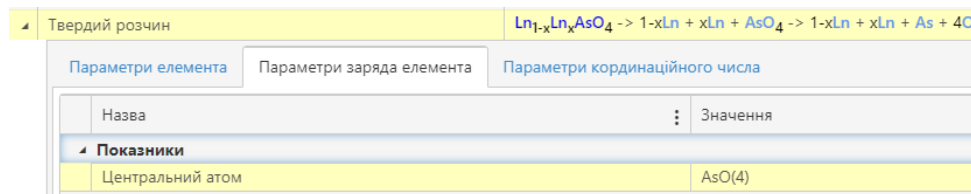


Рисунок 3 – Введення параметрів заряду елемента

3. В процесі створення елемента враховано можливість внесення параметрів координаційного числа до елемента (функція $ElementParamsChargeCoordination$). Так, для кожного елемента потрібно забезпечити отримання показників з бази даних. Отже, якщо параметр pcc_n деякого n -го елемента з K відсутній в системі, обирається універсальне значення pcc_{def} , інакше необхідно ввести значення pcc_n . Формально визначимо продукцію $ParamsChargeCoordinationR$:

Якщо $C(pcc_n)$ тоді $ElementParamsChargeCoordination(pcc_n)$ інакше $ElementParamsChargeCoordination(pcc_{def})$, де предикат $C(pcc_n)$ визначається так:

$$\exists pcc_n \mid n \in \{1, 2, \dots, K\} \rightarrow C(pcc_n) = True$$

На рисунку 4 показано приклад введення параметрів заряду елемента.

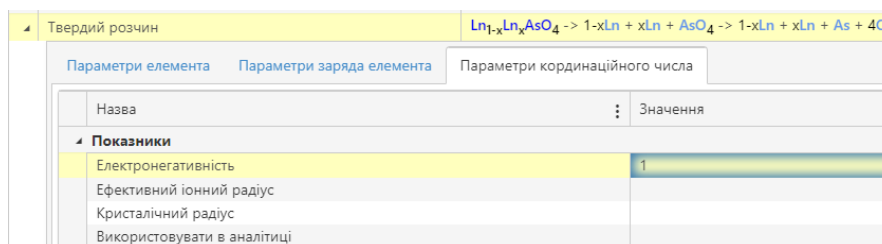


Рисунок 4 – Введення параметрів координаційного числа до елемента

4. Користувачі системи мають змогу самостійно редагувати вид елемента, який їм потрібно. З метою підтримки гнучкості розроблено випадаючий список у вікні елемента: “Складний” та “Простий”. Через меню “Складний” користувачі мають змогу обрати ті j -ті стрічки r_j з LR таблиці елемент, які потрібні для побудови складного елемента (2+ елементів) $ElementTypeAdd$, а через меню “Простий” є змога обирати елементи (до 2-х елементів) у певні j -ті поля з LF таблиці елемент функцією $ElementType$. Формально визначимо продукції $TypeAddR$ та $TypeR$:

Якщо $TypeAdd_j$ тоді $ElementTypeAdd(r_j)$ інакше $ElementTypeAdd(Null)$, де предикат $TypeAdd_j$ визначається так:

$$\exists TypeAdd_j \mid j \in \{1, 2, \dots, LR\} \rightarrow TypeAdd_j = True$$

Якщо $Type_j$ тоді $ElementType(r_j)$ інакше $ElementType(Null)$, де предикат $Type_j$ визначається так:

$$\exists Type_j \mid j \in \{1, 2, \dots, LF\} \rightarrow Type_j = True$$

На рисунку 5 показано приклад зміни виду елемента.



Рисунок 5 – Зміна виду елемента

5. Для того, щоб можна було візуально легко показати, як створився елемент та які елементи занесено до системи, було створено функцію – друк таблиці елементів $PrintElement$. Власне кожен елемент формується з певного набору полів: elm_id (унікальний ід-код певного елемента), elm_selmg (ідентифікатор підгрупи), elm_name (назва елемента), elm_code (хімічний код), elm_count (кількість атомів елемента). Формально оператор друку елемента

$$O_{print} : (elm_id, elm_selmg, elm_name, elm_code, elm_selmg) \rightarrow PrintElement.$$

На рисунку 6 показано приклад друку таблиці елементів

 A screenshot of a spreadsheet application showing a table of elements. The table has columns labeled A through H. The data is as follows:

Id	Id заряду	Id координаційного числа	Назва	Формула	Кількість	Заряд	Координаційне число
42	42	61	Скандій	Sc	1	3	8
42	42	42	Скандій	Sc	1	3	6
101	101	101	Ванадій	V	1	5	6
112	112	112	Арсеніум	As	1	5	6
114	114	114	Иттрий	Y	1	3	6
41	41	41	Лантан	La	1	3	6
22	25	22	Церій	Ce	1	3	6
103	103	103	Празеодин	Pr	1	3	6
104	104	104	Неодим	Nd	1	3	6
105	105	105	Прометий	Pm	1	3	6
21	24	21	Самарій	Sm	1	3	6
61	61	63	Європій	Eu	1	3	6
81	81	81	Гадоліній	Gd	1	3	6
106	106	106	Тербій	Tb	1	3	6
107	107	107	Диспрозій	Dy	1	3	6
108	108	108	Гольмій	Ho	1	3	6
43	43	43	Ербій	Er	1	3	6
43	43	62	Ербій	Er	1	3	8
109	109	109	Тулій	Tm	1	3	6
110	110	110	Иттербий	Yb	1	3	6
111	111	111	Лютецій	Lu	1	3	6
24	22	45	Оксиген	O	4	-2	6

Рисунок 6 – Друк таблиці елементів

Під терміном прогнозування будемо розуміти складову, яка починається з моменту аналізу, обробки даних та закінчуючи виведенням даних на інтерфейс користувача. Використовуючи дані з таблиць: *element*, *charge_element*, *coordination_element*, *volume_cell*, *term_system*, *atom_length*, *sum_atom_length*, *stored_system*, *stored_result*, проаналізувавши їх та обробивши, можливо отримати дані, які буде виведено на інтерфейс користувача у вигляді графіку та таблиці. Для цього розроблено базу знань продукційного типу інтелектуального модуля “Прогнозування”, що складається з множини продукційних правил, предикатів, функцій та операторів. Зокрема:

1. В процесі прогнозування спочатку відбувається аналіз даних (функція *PredAnalysis*). Для кожної системи необхідно забезпечити отримання показників кожного елемента з бази даних та проаналізувати, чи можливо зробити прогнозування. Отже, якщо параметр a_i деякого i -го елемента з AN відсутній в системі, обирається універсальне значення a_{def} , інакше необхідно ввести значення a_i . Формально визначимо продукцію *Analysis*:

Якщо $PA(a_i)$ тоді *PredAnalysis* (a_i) інакше *PredAnalysis* (a_{def}), де предикат $PA(a_i)$ визначається так:

$$\exists a_i \mid i \in \{1, 2, \dots, AN\} \rightarrow PA(a_i) = True$$

2. Для того, щоб можна було візуально легко показати, які помилки виникли при аналізі даних, було створено функцію – виведення помилок *PredAnalysisErr*. Функції необхідно передати певний набір полів: *elm_id* (унікальний ід певного елемента-системи), *strs_id* (ідентифікатор структури розчину), *elm_id_ln* (ід певного лантаніда), *eps* (крок обчислень). Формально оператор виведення помилок при аналізі системи

$$O_{analysis} : (elm_id, strs_id, elm_id_ln, eps) \rightarrow PredAnalysisErr.$$

На рисунку 7 показано приклад виведення помилки при аналізі системи.

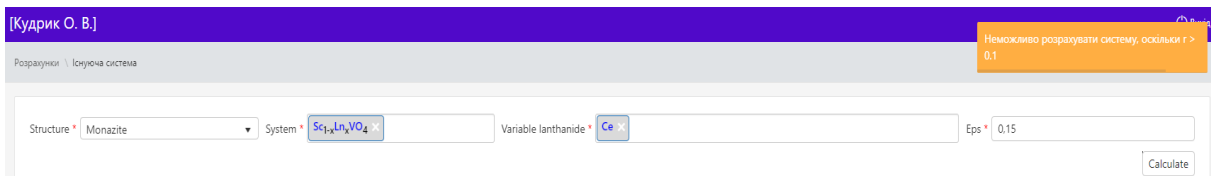


Рисунок 7 – Виведення помилки при аналізі системи

3. Під час аналізу даних, також, перевіряється, чи не була вже така система з тим самим набором вхідних даних розрахована (функція *PredAnalysisData*), якщо була розрахована, то виводимо знайдені дані. Проводиться аналіз та порівняння даних з бази даних, а саме в таблиць: *stored_system* та *stored_result*. Отже, якщо дані s_d деякого d -го елемента з SR наявні в системі, передаємо значення s_d , інакше s_{found} . Формально визначимо продукцію *AnalysisData*:

Якщо $AD(s_d)$ тоді *AnalysisData* (s_d) інакше *AnalysisData* (s_{found}), де предикат $AD(s_d)$ визначається так:

$$\exists s_d \mid d \in \{1, 2, \dots, AD\} \rightarrow AD(s_d) = True$$

4. Також було створено функцію – *PredAnalysisDataCheck*, яка перевіряє, чи раніше розраховувалась система. Функції необхідно передати певний набір полів: *elm_id* (унікальний ід певного елемента-системи), *strs_id* (ідентифікатор структури розчину), *elm_id_ln* (ід певного лантаніда), *eps* (крок обчислень). Формально оператор виведення помилок при аналізі системи

$$O_{dataCheck} : (elm_id, strs_id, elm_id_ln, eps) \rightarrow PredAnalysisDataCheck.$$

5. Після аналізу даних відбувається вибір алгоритма прогнозування (функція *PredTypeAlgorithm*). Система аналізує та порівнює наявність параметрів у елемента з бази даних, до уваги беруться такі параметри: міжатомні довжини, сума міжатомних довжин, об’єм елементарних комірок та іонний радіус. Отже, якщо параметр prm_j деякого j -го елемента з $PARAM$ відсутній в системі, обирається алгоритм з використанням іонних радіусів prm_{ion} , інакше встановлюється алгоритм prm_j . Формально визначимо продукцію *Algorithm*:

Якщо $AL(prm_j)$ тоді *PredTypeAlgorithm* (prm_j) інакше *PredTypeAlgorithm* (prm_{ion}), де предикат $AL(prm_j)$ визначається так:

$$\exists prm_j | j \in \{1, 2, \dots, PARAM\} \rightarrow AL(prm_i) = True$$

6. Для обробки даних, після встановлення типу алгоритма прогнозування, було створено функцію – Prediction. Функції необхідно передати певний набір полів: elm_id (унікальний ід певного елемента-системи), str_id (ідентифікатор структури розчину), elm_id_ln (ід певного лантаноїда), eps (крок обчислень), algorithm_tp (тип алгоритма). Формально оператор прогнозування має вигляд

$$O_{pred} : (elm_id, str_id, elm_id_ln, eps, algorithm_tp) \rightarrow Prediction.$$

7. Після обробки даних, зберігаємо результати в спеціально розроблених таблицях - stored_system та stored_result. Ці таблиці в подальшому будуть використовуватись на етапі аналізу даних, для того, щоб перевірити, чи раніше проводились розрахунки з тими ж самими вхідними даними.

Отже, було створено функцію – SaveResult.

Функції необхідно передати певний набір полів: stds_strs (ідентифікатор структури системи), stds_ln1 (ідентифікатор лантаноїда 1), stds_ln2 (ідентифікатор лантаноїда 2), stds_anion (ідентифікатор аніона), stds_eps (крок обчислень), stdr_stds (ідентифікатор розрахованої системи), stdr_x (результат x), stdr_x1 (результат x1), stdr_x2 (результат x2), stdr_t (результат t), stdr_t_crit (результат t_crit), stdr_q (результат q), stdr_method (результат method), stdr_err_mesage (текст помилки).

Формально оператор прогнозування має вигляд:

$$O_{saveResult} : (stds_strs, stds_ln1, stds_ln2, stds_anion, stds_eps, stdr_stds, stdr_x, stdr_x1, stdr_x2, stdr_t, stdr_t_crit, stdr_q, stdr_method, stdr_err_method) \rightarrow SaveResult.$$

8. Для відображення даних на інтерфейсі користувача, було створено функцію – ReturnResult. Функції необхідно передати певний набір полів: stdr_id (унікальний ід-код даних про розраховану систему). Формально оператор відображення даних на інтерфейсі користувача

$$O_{returnResult} : (stdr_id) \rightarrow ReturnResult.$$

На рисунку 8 показано приклад відображення даних на інтерфейсі користувача

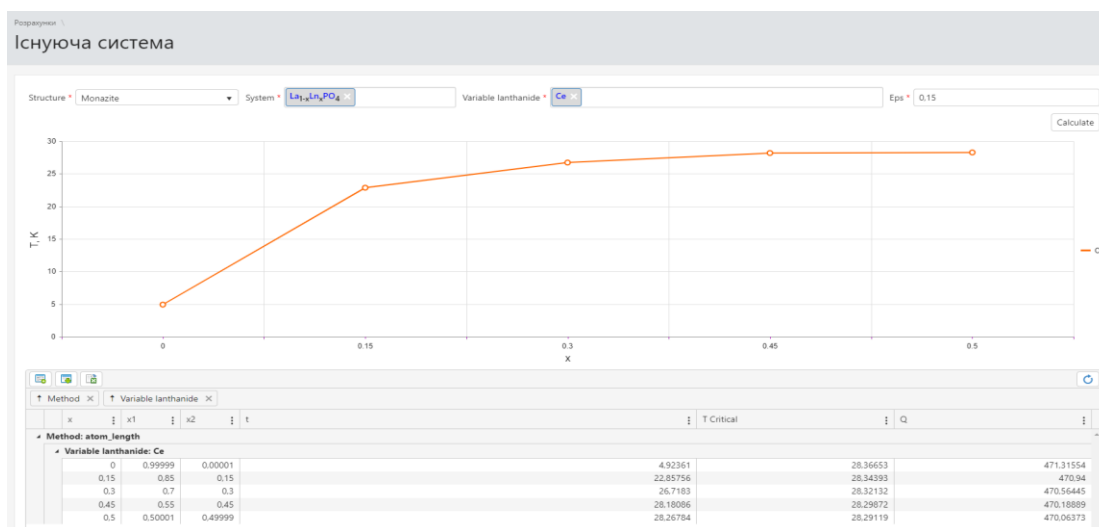


Рисунок 8 – Відображення даних на інтерфейсі користувача

Висновки

В даній роботі вперше створено модель бази знань для прогнозування фазової стабільності твердих розчинів, яка, на відміну від існуючих, враховує унікальну границю заміщень кожного лантаноїду за допомогою множини продукційних правил, предикатів, функцій та операторів, що дозволяє підвищити точність розрахунку енергії змішування та обчислити точну, а не загальну критичну температуру розкладання відповідного лантаноїду.

Список літератури

- [1] Z. Nagy, *Artificial Intelligence and Machine Learning Fundamentals*. Packt Publishing, 330 p., 2018.

- [2] С. О. Субботін, А. О. Олійник, О. О. Олійник, *Неітеративні, еволюційні та мультиагентні метод синтезу нечіткологічних і нейромережних моделей: Монографія*. Запоріжжя: ЗНТУ, 375 с., 2009.
- [3] В. В. Пасічник, Н. Б. Шаховська, *Сховища та простори: Монографія*. Львів: Вид-во Нац. ун-ту «Львівська політехніка», 244 с., 2009.
- [4] П. Джексон, *Введення в експертні системи*. Видавництво Вільямс, 2001. ISBN 5-8459-0150-2, ISBN 0-201-87686-8.
- [5] В. І. Шинкарук, *Логіко-інформаційна система*. Інститут філософії імені Григорія Сковороди НАН України: Абрис, 742 с., 2002.
- [6] O. V. Kudryk O. V. Bisikalo, Yu. A. Oleksii, S. V. Radio, «Intelligent information system for predicting chemicals with interactive possibilities», *CoLInS, Computational Linguistics and Intelligent Systems*. CoLInS 2021. [Електронний ресурс]. Режим доступу: <http://ceur-ws.org/Vol-2870/paper68.pdf>. Дата звернення: 09.02.23.

Стаття надійшла: 09.02.2023.

References

- [1] Z. Nagy, *Artificial Intelligence and Machine Learning Fundamentals*. Packt Publishing, 330 p., 2018.
- [2] S. O. Subbotin, A. O. Oliynyk, O. O. Oliynyk, *Non-iterative, evolutionary and multi-agent methods of synthesis of fuzzy-logical and neural network models: Monograph*. Zaporizhzhia: ZNTU, 375 p., 2009 [in Ukrainian].
- [3] V. V. Pasichnyk, N. B. Shakhovska, *Storage and spaces: Monograph*. Lviv: National Publishing House «Lviv Polytechnic University», 244 p., 2009 [in Ukrainian].
- [4] P. Jackson, *Introduction to Expert Systems*. Williams Publishing, 2001. ISBN 5-8459-0150-2, ISBN 0-201-87686-8 [in Ukrainian].
- [5] V. I. Shinkaruk, *Logic-informational system*. Hryhoriy Skovoroda Institute of Philosophy of the National Academy of Sciences of Ukraine: Outline, 742 p., 2002 [in Ukrainian].
- [6] O. V. Kudryk O. V. Bisikalo, Yu. A. Oleksii, S. V. Radio, Intelligent information system for predicting chemicals with interactive possibilities, *CoLInS, Computational Linguistics and Intelligent Systems*. CoLInS 2021. [Online]. Available: <http://ceur-ws.org/Vol-2870/paper68.pdf>. Accessed on: 09.02.2023.

Відомості про авторів

Бісікало Олег Володимирович – доктор технічних наук, професор, завідувач кафедри автоматизації та інтелектуальних інформаційних технологій.

Кудрик Олексій Володимирович – аспірант групи 126-21а, кафедра автоматизації та інтелектуальних інформаційних технологій.

O. V. Bisikalo, O. V. Kudryk

KNOWLEDGE BASE IN INTELLIGENT INFORMATION SYSTEM FOR PREDICTING OF PHASE STABILITY OF SOLID SOLIDS

Vinnitsia National Technical University, Vinnitsia