

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Державне підприємство «Антонов»
Державний концерн «Укроборонпром»

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Міністерство освіти і науки України

Кваліфікаційна наукова
праця на правах рукопису

Конотоп Дмитро Ігорович

УДК 004.94:005.311.2

ДИСЕРТАЦІЯ

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ СТВОРЕННЯ ТА СУПРОВОДЖЕННЯ УЗАГАЛЬНЕНОЇ МОДЕЛІ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ

05.13.06 – Інформаційні технології

Подається на здобуття наукового ступеня кандидата технічних наук

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело



Д. І. Конотоп

Науковий керівник: Зінченко Валерій Петрович, кандидат технічних наук, доцент

Київ – 2019

Світлій пам'яті
Зінченка Валерія Петровича,
який керував, допомагав,
надихав своїм прикладом.

Світлій пам'яті моїй бабусі,
Конотоп Галини Мусіївни,
справжнього вчителя, патріота,
людини, яка творила тільки добро.

Велика вдячність
«аксакалам» ДП «Антонов»
та кафедрі АЕД
за вдалі поради та цікаві рішення;
та моїй сім'ї за допомогу та підтримку
під час написання роботи.

АНОТАЦІЯ

Конотон Д. І. Інформаційна технологія створення та супроводження узагальненої моделі складних технічних об'єктів. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.06 – Інформаційні технології. – Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, 2019.

В дисертаційній роботі досліджено та розроблено інформаційну технологію (ІТ) створення та супроводження узагальненої моделі (УМ) складних технічних об'єктів (СТО) з метою зниження термінів створення СТО та зменшення їхньої вартості при забезпеченні відповідної якості за рахунок налагодження зв'язку між моделями СТО та поєднання складових інформаційних технологій CALS (підтримка безперервних постачань і життєвого циклу (ЖЦ) та PLM-рішень (керування ЖЦ).

Теоретичні дослідження виконані на підставі методів експертного оцінювання, структурного системного аналізу, теорії множин, нечіткої логіки та теорії алгоритмів – для дослідження та розробки узагальненої моделі, методу знання-орієнтованого створення складних технічних об'єктів та інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів; об'єктно-орієнтованого програмування, баз даних, представлення знань – для вирішення завдання керування параметричною інформацією при побудові моделей складного технічного об'єкта та для обміну даними між складовими інформаційних технологій CALS та PLM-рішень. На основі комплексного аналізу наукових літературних джерел та досвіду практичної роботи з впровадження складових інформаційних технологій CALS та PLM-рішень на Державному

підприємстві (ДП) «Антонов» та інших провідних підприємствах-розробниках СТО, визначено, що:

1. На кожному етапі створення СТО розроблена модель включає велику кількість елементів та міжелементних параметричних виразів, геометричних посилянь, обмежень і т.п., що мають свої алгоритми побудови.

2. Моделі різних етапів створення СТО не повністю пов'язані між собою.

3. Моделі пов'язані з іншими моделями СТО, проте модифікації в моделях, які мають нести навантаження на інші моделі, є величезними і схильними до помилок.

4. Взаємодія складових IT CALS та PLM-рішень складна, що проявляється в проблемах конвертації даних з одних програмних продуктів в інші та веде до часткової або повної втрати історії побудови моделі, топології та параметрів моделі.

Розроблено інформаційну технологію створення та супроводження узагальненої моделі складних технічних об'єктів для вирішення вищезазначених проблем.

Вперше сформовано та досліджено узагальнену модель складних технічних об'єктів, яка дозволяє налагодити інформаційний зв'язок між моделями створення складних технічних об'єктів. Вперше розроблено метод знання-орієнтованого створення складних технічних об'єктів, що дозволяє забезпечити взаємозв'язок між етапами створення складних технічних об'єктів і поєднати складові IT CALS та PLM-рішень. Розроблено модифікацію методу моделювання складних технічних об'єктів на основі знання-орієнтованих технологій для обміну даними між складові IT CALS та PLM-рішень – це процес конвертації даних в різних ІС САх (комп'ютерного проектування, аналізу та виробництва), який отримано після проведеного аналізу методів обміну даними при створенні моделі в різних ІС САх та виявленні їхніх недоліків.

Для реалізації інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів виконано розробку засобів: бази

знань, бази даних, розроблено прикладне програмне забезпечення; обрано складові IT CALS та PLM-рішень.

Розроблено файловою базу даних, що містить параметри для визначення моделей складних технічних об'єктів на прикладі літального апарату, де зберігаються параметри, що описують характеристики, особливості конструювання, економічні параметри літального апарату і т.д. Описано складові бази даних підприємства, що містить моделі складних технічних об'єктів, результати розрахунків, різну службову інформацію та знаходиться в ІС PDM (керування даними про виріб).

Розроблено базу знань побудови моделі складних технічних об'єктів, представлено онтологічну схему розробки моделей складних технічних об'єктів. Онтологія початкових етапів побудови моделей складних технічних об'єктів описує основні зв'язки та співвідношення між частинами процесу побудови моделей. Отримана система класів, підкласів, атрибутів і зв'язків реалізована мовою ontology web language (OWL), функції інтерпретації, і правила вибору, реалізовані за допомогою OWL і Java, складають основу бази знань і системи прийняття рішень, використовуючи знання-орієнтовані технології в моделюванні.

Розроблено систему програмних модулів для знання-орієнтованої побудови моделей СТО, що створює швидкі та якісні умови для внесення змін в процес проектування та виробництва СТО. Середовище програмних модулів за допомогою спеціально налаштованих дій модулів здатне весь час підтримувати дуплексний зв'язок із базою даних і дозволяє: робити запит та отримувати інформацію про складові, правила, умови та обмеження бази знань.

Після проведеного огляду обчислювального середовища та урахування світового досвіду провідних компаній-виробників СТО, прийняте рішення використовувати системи САХ: NX та CATIA під загальним керуванням системи PLM TeamCenter; База даних ведеться в MySQL за допомогою системи керування баз даних DbVisualizer; для розробки бази знань використовуються програми Protege 4.3 та DbVisualizer; в роботі обране середовище Visual Basic для зв'язку

програми САПІА з базою даних; і середовище Java для розробки системи програмних модулів та для зв'язку програми NX з базою даних.

Реалізовано знання-орієнтоване середовище керування параметричною інформацією шляхом використання онтології для опису зв'язку, обмежень та правил між складовими процесу створення моделей СТО. Реалізовано зв'язок знання-орієнтованого керування параметрами моделей за допомогою програмних модулів.

Розробка нового методу створення СТО здійснюється за допомогою середовища Java, в якому програмні модулі використовують інформацію про параметри моделей з бази знань. Наведено приклад розв'язку проектної задачі зв'язку моделей та параметрів бази даних на прикладі побудови моделей вантажного люку літального апарату.

Наукова новизна роботи полягає в наступних основних положеннях.

Вперше запропоновано узагальнену модель складних технічних об'єктів, яка на основі використання теоретико-множинного підходу дозволяє налагодити інформаційний зв'язок між моделями процесу створення складних технічних об'єктів.

Вперше розроблено метод знання-орієнтованого створення складних технічних об'єктів на базі узагальненої моделі, що забезпечує взаємозв'язок між етапами створення складних технічних об'єктів, а також дозволяє поєднати складові інформаційних технологій CALS та PLM-рішень.

Удосконалено метод інтегрованого проектування та комп'ютерного моделювання за рахунок введення рівня керування параметрами моделей складних технічних об'єктів з використанням бази даних та знань, що дозволяє розширити функціональні можливості методу та зробити його більш гнучким.

Розроблено інформаційну технологію створення та супроводження узагальненої моделі складних технічних об'єктів на базі методу знання-орієнтованого створення складних технічних об'єктів, що дозволило: автоматизувати процес обробки та побудови моделей; доповнити технологію

паралельного проектування PLM та складові інформаційних технологій CALS та PLM-рішень для завдання створення моделей складних технічних об'єктів.

Практичне значення одержаних результатів полягає у розробці засобів забезпечення реалізації інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів: бази даних "Параметри моделей", що ведеться в MySQL за допомогою системи керування базами даних DbVisualizer; бази знань, використовуючи програми Protege 4.3 та DbVisualizer, в якій зберігаються обмеження на складові бази даних відповідно вимогам нормативних документів і т. д., що створює швидкі та якісні умови для внесення змін в процес проектування і виробництва складних технічних об'єктів за допомогою спеціально налаштованих дій програмних модулів, розроблених в середовищі Java. Розроблено програмні модулі для зв'язку з базою даних "Параметри моделей": інформаційної системи CAx NX в середовищі Java; та інформаційної системи САТІА в середовище Visual Basic. Розроблено інформаційну систему створення та супроводження узагальненої моделі складних технічних об'єктів, в якій розроблені інструментальні засоби приєднано до наявних серверів підприємства, використовуючи інформаційну систему PLM TeamCenter, з яких отримуються дані, що використовується в процесі створення складних технічних об'єктів.

Розроблені автором засоби інформаційної технології використані в процесі створення моделей літальних апаратів на ДП "Антонов". Реалізація інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів при проектуванні легкого транспортного літака та турбогвинтового регіонального вантажопасажирського літака на прикладі моделей вантажного люку забезпечує близько 22% економії матеріальних витрат. Основні положення та висновки дисертаційної роботи використані в навчальному процесі КПІ ім. Сікорського і Національного університету кораблебудування імені адмірала Макарова. Є відповідні акти впровадження.

Впровадження інформаційної технології створення та супроводження узагальненої моделі СТО доповнює технології паралельного проектування ІС

PLM та складові IT CALS та PLM-рішень для завдання створення моделей СТО, що забезпечує зменшення термінів створення СТО за рахунок керування параметрами моделі СТО. Наприклад, порівнюємо час розробки усіх етапів побудови моделей ВЛ в ІС САх та PLM: наявними ІТ відповідно план-графіку час складає 7280 людино-годин, а розробленою ІТ створення та супроводження узагальненої моделі СТО – 5660, що показала близько 22% економії матеріальних витрат. І це тільки при розгляді аспекту застосування модифікації методу знання-орієнтованого створення СТО для керування параметричною інформацією. У випадку проектної кооперації, коли використовується декілька ІС САх, запропонована модифікація методу знання-орієнтованого створення складного технічного об'єкта для обміну даними між різними складовими ІТ CALS та PLM-рішень, надасть додаткову ефективність використання ІТ за рахунок того, що ведеться лише одна БЗ та БД – спільна для всіх використаних ІС САх та PLM.

Ключові слова: інформаційна технологія, інформаційна система, узагальнена модель, складний технічний об'єкт, знання-орієнтована технологія, CALS, PLM, база даних, база знань.

Основні результати дисертаційної роботи опубліковані в 24 наукових працях.

ANNOTATION

Konotop D. I. Information technology of complex technical objects common model creation and support. - Qualifying scientific work on the right of manuscript.

The thesis for the degree of Candidate of Technical Sciences, specialty 05.13.06 – Information technologies. – National Technical University of Ukraine «Kyiv Polytechnic Institute named Igor Sikorskiy», Kyiv, 2019.

Information technology of complex technical objects common model creation and support has been studied and developed with the aim of complex technical objects creation terms cutting and reduction of it prices in the proper quality denotation for providing links between the stages of building models of complex technical objects at different stages of creating a complex technical object and transferring data between the components of information technology CALS and PLM solutions.

Theoretical researches are made on the basis of systems analysis approaches, set theory and the theory of algorithms to research and describe the tasks and approach for modeling of CTO based on knowledge-based technologies. The implementation of method for solving the parametric information management in the construction of models of CTO and to exchange data between different components of CALS technology occur using methods of object-oriented programming, database, knowledge representation. Based on a comprehensive analysis of scientific literature and practical experience with the implementation of CALS technology components at the "Antonov" company and other leading companies of CTO design, stipulates that:

1. at each stage of CTO design developed model includes many elements and connections between elements parametric expressions, geometric references, limitations, etc., have their own algorithms.

2. models of various design phases of CTO are not fully connected.

3. models associated with other models of CTO, but modifications in models which have to bear the burden on other models, is huge and prone to errors.

4. Interaction components of CALS is complex, which manifests itself in problems converting data from one software to another and leads to partial or complete loss of construction history of model, topology and parameters of models.

The approach of modelling of complex technical objects using knowledge-based technology is developed to solve these problems.

The algorithm of modelling of complex technical objects using knowledge-based technology to control of the parametric information of models creation to solve problems tracking the identity of the source data at different stages of building a model of CTO is provided for the establishment of the relationship stages of building models CTO.

The algorithm of modelling of complex technical objects using knowledge-based technology to data communication between different parts of CALS technologies is designed. This is the process of data converting at different computer-aided (CAx) systems which is received after analyze of data transferring of models from different CAx and their disadvantages were defined.

To implement the method of modeling of complex technical objects based on knowledge-oriented technologies the means are developed: knowledge base, database; application software is developed; components of CALS are selected.

Database that contains an option to specify models of complex technical objects on the example of the aircraft is developed, which stores parameters that describe the characteristics, layout, economic parameters of the aircraft, etc. The components of the enterprise database containing models of complex technical objects, calculation results, and various service information stored in the product lifecycle management (PLM, product lifecycle management) are described.

The knowledge base for constructing models of complex technical objects is developed; ontological scheme of development of models of complex technical projects is presented. Ontology of initial stages of construction of models of complex technical objects describes the basic connections and relationship between parts of the process of building models. The resulting system of classes, subclasses, attributes and relationships is implemented by ontology web language (OWL), features of interpretations and rules

for choosing implemented using OWL and Java are the basis of knowledge base and decision-making system, using knowledge-based technology in modeling.

The program modules system for knowledge-based building of models of CTO is designed that creating a fast and qualitative terms to make changes in the design and production of CTO. The program modules environment through customized software modules capable of action at all times maintain a duplex connection with a database and allows: to request and receive information about the components, rules, conditions and limitations of the knowledge base.

After a review of the computing environment and the world experience leading manufacturers of CTO, decided to use the system CAx: NX and CATIA under the overall control system PLM TeamCenter; The database is in MySQL using the database management system DbVisualizer; to develop the knowledge base used by the program Protege 4.3 and DbVisualizer; Visual Basic environment is used for communication of CATIA applications with database; and Java environment is used for program modules development and communications of software of NX with database.

The knowledge-based environment of parametric data control using ontology to describe communication restrictions and regulations between the constituent processes of building models of CTO is implemented. The communication of knowledge-oriented management of models parameters with program modules is implemented.

Development of a new method of CTO designing is occurred by using Java environment, in which program modules use information about models from the knowledge base. The example of problem solving the communication of models and parameters from database is applied on the example of building a model cargo hatch of the aircraft.

Scientific novelty consists in the following main positions.

For the first time the method of modeling of complex technical objects based on knowledge-based technologies is developed. For the first time the algorithm of method for modeling of complex technical objects on the basis of knowledge-based technologies to manage of parametric information and the algorithm of method for modeling of complex technical objects based on knowledge-based technology for

exchanging data between different components technologies of CALS – is a process of data convert in various CAx are developed. The technology of basic control structure is improved. The automated processing system and build of models of complex technical objects and components CALS technology to problems of building models of complex technical objects are further developed.

The practical significance of the results is that the thesis proposed the approach and means brought to the engineering level and used in the development of components of CALS technology of complex technical object of industrial and academic applications:

1. The process of models building, which increases the efficiency of designing complex technical subject is improved.

2. The algorithm of method for modeling of complex technical objects on the basis of knowledge-based technologies to manage information and parametric algorithm of method for modeling of complex technical objects based on knowledge-based technology for exchanging data between different components of CALS technologies are developed.

3. Means for modeling of complex technical objects based on knowledge-oriented technologies are developed and the ontological scheme of development of models as a whole and its constituent parts in the previous stages of design is presented. On this basis, a set of programs to ensure the implementation of improved construction of models of complex technical objects an example of aircraft design are developed.

Approach and means used in the process of building models of complex technical facility at SE "Antonov" is designed by the author. Implementation of the method of modeling of complex technical objects based on knowledge-oriented technologies in the design of AN-178 and AN-132 provides approximately 22% savings in material costs. The main provisions and conclusions of the thesis used in the educational process of the National Technical University of Ukraine "Kyiv Polytechnic Institute named Igor Sikorsky" and the National University of Shipbuilding named Admiral Makarov.

The efficiency of the method used modeling of complex technical objects based on knowledge-oriented technologies, solving design challenges is researched. Implementation of method of modeling of complex technical objects based on

knowledge-oriented technology is a significant improvement technology of basic control structure and components of CALS technology to problems of building models of complex technical objects, providing the shortening design through control parameters of model. For example, while all stages of building models of aircraft cargo hatch system NX: existing methods in accordance with schedule time is 7280 man-hours and developed by modeling of complex technical objects based on knowledge-oriented technologies - 5660, which gives approximately 22% savings in material costs. And it is only by considering aspects of the algorithm method for modeling of complex technical objects on the basis of knowledge-based technologies to manage parametric information. If the company uses other CAx systems, then the algorithm method for modeling of complex technical objects based on knowledge-based technology for exchanging data between different components of CALS technology provides additional efficiency by a method that is only one knowledge base and the base data - common to all systems used CAx.

Keywords: information technology, information system, common model, complex technical object, knowledge-based technology, CALS technology, PLM, data base, knowledge base.

The main results of dissertation were published in 24 scientific works.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у наукових фахових виданнях

1. Зинченко В. П. Анализ средств и методов информационных технологий синтеза структур конечно-элементных моделей / В. П. Зинченко, В. В. Борисов, Д. И. Конотоп // Інформаційні системи, механіка та керування. – 2009. – № 3. – С. 112–121. *(автору належить: аналіз проблеми взаємодії моделей та ІС PLM.)*
2. Зінченко В. П. Концепція застосування інтелектуальних технологій в проектуванні / В. П. Зінченко, Д. І. Конотоп, Г. В. Деркач, Є. Ю. Абрамов // Открытые информационные и компьютерные интегрированные технологии. – Харьков: Гос. Аэроком. Ун-т “ХАИ”, 2011. – № 49. – С. 169–179. *(автору належить: розкриття методу знання-орієнтованого створення складних технічних об'єктів та його модифікації.)*
3. Зінченко В. П. Інформаційні технології моделювання компоновки складного технічного об'єкта / В. П. Зінченко, Д. І. Конотоп, О. П. Сидоренко, В. В. Борисов // Інформаційні системи, механіка та керування. – 2011. – № 6. – С. 27–35. *(автору належить: формалізація проблематики розробки інформаційних технологій процесу побудови моделей.)*
4. Abramova A. V. Ontology application for estimation of complex technical object characteristics / A. V. Abramova, D. I. Konotop // Електроніка та системи управління. – 2012. – №3 (33). – С. 81–88. *(автору належить: опис методу та засобів знання-орієнтованого створення складних технічних об'єктів; його модифікацій.)*
5. Конотоп Д. І. Створення контрольної базової моделі в інформаційних технологіях проектування складного технічного об'єкта / Д. І. Конотоп, В. П. Зінченко // Наукові вісті Національного технічного університету України "Київський політехнічний інститут". – 2012. – № 6 (86). – С. 132–137. *(автору належить: розробка модифікації методу знання-орієнтованого створення*

складних технічних об'єктів для керування параметричною інформацією при створенні моделей.)

6. Абрамова Г. В. Алгоритми планування вагових розрахунків літака / Г. В. Абрамова, Д. І. Конотоп // Вісник Національного Авіаційного Університету. – 2013. – № 2 (55). – С. 129–135. *(автору належить: розробка методу та модифікації методу знання-орієнтованого створення складних технічних об'єктів.)*

Стаття у виданні, що входить до наукометричної бази Index Copernicus

7. Konotop D. I. 3D-models design concept of complex technical objects using knowledge-based technology/ D. I. Konotop, V. P. Zinchenko // Механіка гіроскопічних систем. – 2017. – № 34. – С. 5–13. *(автору належить: опис методу знання-орієнтованого створення складних технічних об'єктів та його модифікацій.)*

Стаття у іноземному виданні

8. Конотоп Д. И. Оптимальное проектирование сложных технических объектов с использованием онтологического подхода / Д. И. Конотоп, В. П. Зинченко // Научный журнал "Онтология проектирования". – Самара, "Новая техника". – 2011. – № 1(2). – С. 44–53. *(автору належить: опис методу знання-орієнтованого створення складних технічних об'єктів та його модифікацій.)*

Матеріали науково-технічних конференцій

9. Зінченко В. П. Проблеми оптимізації компоновки складного технічного об'єкту / В. П. Зінченко, С. В. Зінченко, Д. І. Конотоп, В. В. Борисов. // II наук. конференція магістрів та аспірантів, присвячена 20-річчю факультету прикладної

математики (Київ, 14 - 16 квітня 2010 р.) – К.: НТУУ “КПІ” – 2010. – Тези доповідей. – С. 64–67.

10. Зінченко В. П. Застосування сучасних комп'ютерних інформаційних технологій при проектуванні складного технічного об'єкту / В. П. Зінченко, Д. І. Конотоп // Міжнародна науково-практична конференція "Інформаційні технології та комп'ютерна інженерія" (Вінниця, 19 - 21 травня 2010 р.) – Вінниця: ВНТУ – 2010. – Тези доповідей. – С. 50–51.

11. Konotop D. Multi-agent-based conception of modern aircraft design / D. Konotop, I. Budinska, V. Zinchenko, E. Gatial // Proceedings of 5th Workshop on Intelligent and Knowledge Oriented Technologies (Bratislava, Slovakia, November 11 - 12, 2010) – p. 125–128.

12. Абрамов Є. Ю. Алгоритмізація бази знань проектування складного технічного об'єкту / Є. Ю. Абрамов, Д. І. Конотоп, Г. В. Деркач // XVIII Міжнародна конференція з автоматичного управління “Автоматика/Automatics – 2011” (Львів, 28 - 30 вересня 2011 р.) – Львів: Видавництво Львівської політехніки – 2011 – Матеріали конференції. – С. 358–359.

13. Конотоп Д. І. Визначення керуючої моделі проектування складного технічного об'єкта / Д. І. Конотоп // XII Міжнародна науково-практична конференція молодих учених і студентів «Політ. Сучасні проблеми науки» (Київ, 4 - 5 квітня 2012 р.) – К.: НАУ – 2012. – Тези доповідей. – С. 69.

14. Конотоп Д. И. Использование базовой управляющей структуры в процессе проектирования сложного технического объекта / Д. И. Конотоп // XIX Международная научная конференция студентов, аспирантов и молодых ученых «ЛОМОНОСОВ-2012» (Москва, МГУ имени М. В. Ломоносова, 9 - 13 апреля 2012 г.) – М.: Издательский отдел факультета ВМК МГУ – 2012. – Сборник тезисов. – С. 29–30.

15. Abramova A. V. Application of Sheynin's plural calculations method at the initial stages of complex technical object design / A. V. Abramova, D. I. Konotop // 2-nd International Conference «Methods and Systems of Navigation and Motion Control»

(Kyiv, October, 9 - 12, 2012) – К.: Освіта України – 2012. – Proceedings. – С. 129–132. **(Наукометрична база SCOPUS)**

16. Abramova A. V. Using graphs for the planning at the initial stages of complex technical object design / A. V. Abramova, D. I. Konotop // The 2nd International Scientific Conference of Students and Young Scientists "Theoretical and Applied Aspects of Cybernetics" (Kyiv, November, 12 - 16, 2012) – К.: "Bukrek" – 2012. – Proceedings. – С. 8–13.

17. Конотоп Д. І. Застосування онтології в обміні даними при розробці складного технічного об'єкта / Д. І. Конотоп // Збірник доповідей ІХ Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки”. – Київ, 17 - 18 квітня 2013 – Секція "Інформаційні технології". – С. 437–440.

18. Abramov E. Knowledge-oriented support of complex technical object design / E. Abramov, D. Konotop, A. Abramova // 2-nd International Conference «Actual problems of UAV development» (Kyiv, October, 15 - 17, 2013) – Proceedings. – P. 122–125. **(Наукометрична база SCOPUS)**

19. Konotop D. Using ontology in geometrical modeling of complex technical object. / D. Konotop // VII International students and young scientists conference «Intelligence. Integration. Reliability» (Kyiv-Warsaw, April, 28 - 29, 2014) – К.: ІВЦ "Видавництво «Політехніка»" – 2014 – Abstracts. – P. 20–21.

20. Konotop D. Ontology using in geometrical models data processing of complex technical object / D. Konotop // XX-th International Conference «Knowledge-Dialogue-Solution» (Kyiv, September, 8 - 10, 2014) – ІТНЕА. Kyiv - Sofia – 2014 – Proceedings. – P. 118–119.

21. Конотоп Д. І. Використання знання-орієнтованих систем при створенні складного технічного об'єкта / Д. І. Конотоп, В. П. Зінченко, Лі Вей // Збірник доповідей Х Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки” (Київ, 16 - 17 квітня 2015) – Секція "Інформаційні технології". – С. 206–211.

22. Конотоп Д. І. Знання-орієнтована побудова геометричних моделей складних технічних об'єктів / Д. І. Конотоп, В. П. Зінченко // Збірник доповідей XI Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки” (Київ, 13 - 14 квітня 2017) – Секція "Інформаційні технології". – С. 72–75.

23. Конотоп Д. І. Створення та супроводження узагальненої моделі складних технічних об'єктів / Д. І. Конотоп // Збірник доповідей IX Міжнародної науково-технічної конференції “Інновації в суднобудуванні та океанотехніці” (Миколаїв, 18 - 19 жовтня 2018) – Секція "Інформаційні технології та управління проектами в промисловості, освіті та соціальній сфері". – С. 402–405.

Навчальний посібник

24. Зінченко В. П. Обчислювальні мережі: побудова та програмування: навч. посібник / В. П. Зінченко, С. В. Зінченко, І. В. Мірошніченко, Д. І. Конотоп, В. Резаї – К. : НТУУ «КПІ», 2011. – 240 с.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	21
ВСТУП.....	23
РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ТА СУПРОВОДЖЕННЯ МОДЕЛЕЙ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ.....	31
1.1 Сутність наявного процесу створення складних технічних об'єктів.....	31
1.2 Проблемні задачі процесу створення складних технічних об'єктів.....	36
1.3 Огляд наявних інформаційних технологій забезпечення життєвого циклу складних технічних об'єктів.....	42
1.4 Постановка задачі розробки інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів	51
1.5 Висновки.....	54
РОЗДІЛ 2. РОЗРОБКА УЗАГАЛЬНЕНОЇ МОДЕЛІ ТА МЕТОДУ ЗНАННЯ- ОРІЄНТОВАНОГО МОДЕЛЮВАННЯ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ.....	55
2.1 Концепція інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів.....	55
2.2 Розробка узагальненої моделі складних технічних об'єктів.....	61
2.3 Математичне забезпечення процесу створення узагальненої моделі складних технічних об'єктів.....	65
2.4 Розробка методу знання-орієнтованого створення складних технічних об'єктів.....	76
2.5 Висновки.....	85
РОЗДІЛ 3. РОЗРОБКА ЗАСОБІВ ЗАБЕЗПЕЧЕННЯ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ТА СУПРОВОДЖЕННЯ УЗАГАЛЬНЕНОЇ МОДЕЛІ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ.....	86

3.1 Модифікації методу знання-орієнтованого створення складних технічних об'єктів.....	86
3.2 Онтологічний опис процесу створення узагальненої моделі складних технічних об'єктів.....	93
3.3 Розробка бази даних та бази знань створення узагальненої моделі складних технічних об'єктів.....	100
3.4 Розробка системи програмних модулів зв'язку складових інформаційної технології.....	110
3.5 Висновки.....	114
РОЗДІЛ 4. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ТА СУПРОВОДЖЕННЯ УЗАГАЛЬНЕНОЇ МОДЕЛІ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ.....	117
4.1 Структурна схема інформаційної системи створення та супроводження узагальненої моделі складних технічних об'єктів.....	117
4.2 Реалізація бази даних та бази знань створення узагальненої моделі складних технічних об'єктів.....	121
4.3 Реалізація системи програмних модулів зв'язку складових інформаційної технології.....	132
4.4 Приклад реалізації інформаційної технології з наявними компонентами CALS та оцінка її ефективності.....	137
4.5 Висновки.....	151
ОСНОВНІ РЕЗУЛЬТАТИ І ВИСНОВКИ.....	152
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	154
ДОДАТКИ.....	175
Додаток А. Схеми створення складних технічних об'єктів.....	176
Додаток Б. Варіанти реалізації інформаційної технології.....	182
Додаток В. Лістинг програм.....	199
Додаток Г. Акти впровадження.....	216
Додаток Д. Список опублікованих праць за темою дисертації.....	222

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

- БД – база даних;
- БЗ – база знань;
- БКС – базова контрольна структура;
- ГМ – геометрична модель;
- ЗОТ – знання-орієнтовані технології;
- ІС – інформаційна система;
- ІТ – інформаційна технологія;
- ЖЦ – життєвий цикл;
- КМ – комп'ютерна мережа;
- ММГ – модель майстер-геометрії;
- МРО – модель розподілу об'єктів;
- МПВ – модель повного визначення;
- ОС – операційна система;
- ПЗ – програмне забезпечення;
- ПК – персональний комп'ютер;
- СКБД – система керування БД;
- СТО – складний технічний об'єкт;
- ТВ – технічні вимоги;
- ТП – технічна пропозиція;
- УМ – узагальнена модель;
- ЧПК – числове програмне керування;
- CALS – Continuous Acquisition and Lifecycle Support (підтримка безперервних постачань і життєвого циклу);
- CASE – Computer-Aided Software Engineering (автоматизація процесів проектування та розробки програмного забезпечення);
- CAx – Computer-Aided systems (комп'ютерне проектування, аналіз та виготовлення);

IDEF – Integrated computer aided manufacturing DEFinition (методологія функціонального моделювання);

PDM – Product Data Management (керування даними про виріб);

PLM – Product Lifecycle Management (керування життєвим циклом виробу);

UML – Unified Modeling Language (уніфікована мова моделювання).

ВСТУП

Актуальність теми

Створення складних технічних об'єктів (СТО, як-от об'єкти наукоємного машинобудування, котрі характеризуються кількістю елементів та зв'язків, що дорівнює 10^6 і більше) – процес, що містить тривалі підпроцеси, складні об'єкти та моделі і спирається на наявні стандарти та інформаційні технології (ІТ), серед яких істотне місце займають: ІТ CALS (Continuous Acquisition and Lifecycle Support – підтримка безперервних постачань і життєвого циклу (ЖЦ) та PLM-рішення (Product Lifecycle Management – керування ЖЦ) СТО. При розробці та дослідженні моделей, які відображають різні сторони структури та функціонування СТО, використовуючи наявні ІТ CALS та PLM-рішення, залишаються невирішеними надзвичайно актуальні проблемні задачі: із забезпечення зв'язків між моделями та етапами створення СТО, поєднання складових ІТ CALS та PLM-рішень, вирішення яких дозволить скоротити тривалість створення СТО та зменшити їхню вартість при забезпеченні відповідної якості.

Основний внесок у розвиток принципів ІТ CALS та PLM-рішень, що пов'язується з необхідністю підвищення якості системних рішень, які готуються і приймаються в результаті функціонування відповідних автоматизованих систем, зробили: В. М. Глушков, М. З. Згуровський, В. С. Михалевич, О. А. Павлов, О. І. Ролік, І. В. Сергієнко, В. І. Скурихін, Н. А. Соколова, С. Ф. Теленик, та інші. Дослідженню завдань розробки моделей СТО з використанням ІТ CALS та PLM-рішень із додаванням проблемно-орієнтованих комплексів і систем приділяли увагу Ю. В. Абрамов, С. А. Бичков, В. А. Богуслаєв, А. Г. Братухін, О. Г. Гребеников, В. П. Зінченко, Г. О. Кривов, П. М. Павленко, М. Т. Фісун та інші вчені. Роботи Т. Грубера, Н. Гуаріно, О. В. Палагіна, М. Стокса визначили ЖЦ обробки знань при розробці СТО і є відправною точкою для використання знань, що містяться в сучасних ІТ CALS та PLM-рішеннях.

На даний момент питаннями впровадження складових IT CALS та PLM-рішень при створенні СТО займаються провідні компанії світу, як-от: Boeing, Airbus, ДП «Антонов», Dymler, General Motors, Ford Motors, Hyundai Heavy Industries та інші; в цьому процесі безпосередньо беруть участь також провідні вищі навчальні заклади зі своїми лабораторіями, зокрема в Україні: Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Сікорського), Національний аерокосмічний університет ім. М.Є. Жуковського "Харківський авіаційний інститут", Національний університет кораблебудування імені адмірала Макарова, Національний авіаційний університет та ін. Основними напрямками їхніх робіт є: розробка загальної методології керування даними у проектах по розробці СТО, автоматизація етапів проектування та виробництва різноманітних моделей СТО, удосконалення методів і засобів проектних досліджень, у тому числі створення проблемно-орієнтованих IT, які дозволяють інтегрувати отримані в процесі досліджень знання, забезпечуючи високу якість досліджень.

З практики відомо, що створені за допомогою інформаційних систем (IS) – складових CALS та PLM-рішень моделі СТО мають наступні недоліки: моделі на різних етапах створення СТО не повністю пов'язані між собою; моделювання СТО відбувається, використовуючи різні складові IT CALS та PLM-рішень що створює постійні складнощі конвертації даних та веде до часткової або повної втрати даних моделі; відсутній автоматизований зв'язок з іншими моделями СТО.

У зв'язку з чим, спостерігається тенденція удосконалення керування IT CALS та PLM-рішень. Зокрема, розроблено проект МОКА (Managing Engineering Knowledge – керування інженерними знаннями) на основі знання-орієнтованих технологій (ЗОТ). Аналіз складових IT CALS та PLM-рішень, де базовими є IC PDM (Product Data Management – керування даними про виріб) та САх (Computer-Aided systems – комп'ютерне проектування, аналіз та виготовлення) показав, що бази знань (БЗ) розміщуються безпосередньо в IC САх, що унеможлиблює їхнє використання в інших складових IT CALS та PLM-рішень.

Сучасна практика показує, що ефект від використання різноманітних ІТ при створенні СТО можливий за умови реалізації її раціональної структури за рахунок повноцінного інформаційного, методичного, програмного, технічного, організаційного забезпечення компанії, де створюється СТО на всіх етапах його життєвого циклу з урахуванням вимог єдності й інтеграції даних.

Дослідження, пов'язані з розробкою ІТ створення та супроводження узагальненої моделі СТО в умовах зміни параметрів та обмежень моделей, доповнюють наявні складові ІТ CALS та PLM-рішень і є надзвичайно актуальною науково-практичною задачею.

Зв'язок роботи з науковими програмами, планами, темами

Робота виконувалася на Державному підприємстві «Антонов» (ДП «Антонов») відповідно національної програми створення близькомагістрального транспортного турбореактивного літака АН-178 (наказ № 1967 від 27.08.2010 на ескізне проектування літака АН-178 та Директивний графік створення АН-178 від 28.12.2011) та турбогвинтового регіонального вантажопасажирського літака АН-132 (Директивний графік створення літака-демонстратора АН-132 від 28.04.2015 та наказ № 96 від 10.06.2015 про випуск конструкторської документації). Робота виконана також згідно з планом наукових досліджень кафедри автоматизації експериментальних досліджень Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Сікорського) та в рамках держбюджетної теми 2358 "Розробка засад створення нано- та пікосупутників як космічних мікролабораторій" (R/K/ 0110U002593).

Мета та задачі дослідження

Метою дисертаційної роботи є скорочення тривалості створення СТО та зменшення їхньої вартості при забезпеченні відповідної якості за рахунок налагодження зв'язку між моделями складних технічних об'єктів та поєднання складових інформаційних технологій CALS та PLM-рішень шляхом дослідження та розробки інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів.

Для досягнення поставленої мети необхідно розв'язати основні задачі:

1. Провести системний аналіз наявних технологій, методів та засобів створення моделей складних технічних об'єктів та інформаційних технологій CALS та PLM-рішень.

2. Сформулювати та дослідити узагальнену модель складних технічних об'єктів, яка дозволить налагодити інформаційний зв'язок між моделями створення складних технічних об'єктів.

3. Розробити метод знання-орієнтованого створення складних технічних об'єктів на базі узагальненої моделі, що дозволить забезпечити взаємозв'язок між етапами створення складних технічних об'єктів, а також поєднати складові інформаційних технологій CALS та PLM-рішень.

4. Створити модифікації методу знання-орієнтованого створення складних технічних об'єктів для: керування параметричною інформацією при створенні моделей складних технічних об'єктів; обміну даними між складовими інформаційних технологій CALS та PLM-рішень.

5. Розробити інформаційну технологію створення та супроводження узагальненої моделі складних технічних об'єктів та оцінити її ефективність.

Об'єктом дослідження є інформаційні процеси підтримки етапів життєвого циклу складних технічних об'єктів.

Предметом дослідження є інформаційна технологія створення та супроводження узагальненої моделі складних технічних об'єктів.

Методи дослідження: методи експертного оцінювання, структурного системного аналізу, теорії множин, нечіткої логіки та теорії алгоритмів – для дослідження та розробки узагальненої моделі, методу знання-орієнтованого створення складних технічних об'єктів та інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів; об'єктно-орієнтованого програмування, баз даних, представлення знань – для вирішення завдання керування параметричною інформацією при побудові моделей складного технічного об'єкта та для обміну даними між складовими інформаційних технологій CALS та PLM-рішень.

Інформаційну базу дослідження склали: практичні роботи із впровадження складових інформаційних технологій CALS та PLM-рішень на ДП «Антонов» та інших провідних підприємствах-розробниках СТО; наукові праці провідних вітчизняних і зарубіжних вчених; результати проведених автором наукових досліджень.

Наукова новизна отриманих результатів

Вперше:

– запропоновано узагальнену модель складних технічних об'єктів, яка на основі використання теоретико-множинного підходу дозволяє налагодити інформаційний зв'язок між моделями процесу створення складних технічних об'єктів.

– розроблено метод знання-орієнтованого створення складних технічних об'єктів на базі узагальненої моделі, що забезпечує взаємозв'язок між етапами створення складних технічних об'єктів, а також дозволяє поєднати складові інформаційних технологій CALS та PLM-рішень.

Удосконалено метод інтегрованого проектування та комп'ютерного моделювання за рахунок введення рівня керування параметрами моделей складних технічних об'єктів з використанням бази даних та знань, що дозволяє розширити функціональні можливості методу та зробити його більш гнучким.

Розроблено інформаційну технологію створення та супроводження узагальненої моделі складних технічних об'єктів на базі методу знання-орієнтованого створення складних технічних об'єктів, що дозволило: автоматизувати процес обробки та побудови моделей; доповнити технологію паралельного проектування PLM та складові інформаційних технологій CALS та PLM-рішень для завдання створення моделей складних технічних об'єктів.

Практичне значення отриманих результатів полягає у розробці засобів забезпечення реалізації інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів: бази даних "Параметри моделей", що ведеться в MySQL за допомогою системи керування базами даних DbVisualizer; бази знань, використовуючи програми Protege 4.3 та DbVisualizer, в

якій зберігаються обмеження на складові бази даних відповідно вимогам нормативних документів і т. д., що створює швидкі та якісні умови для внесення змін в процес проектування і виробництва складних технічних об'єктів за допомогою спеціально налаштованих дій програмних модулів, розроблених в середовищі Java. Розроблено програмні модулі для зв'язку з базою даних "Параметри моделей": інформаційної системи CAx NX в середовищі Java; та інформаційної системи САТІА в середовище Visual Basic. Розроблено інформаційну систему створення та супроводження узагальненої моделі складних технічних об'єктів, в якій розроблені інструментальні засоби приєднано до наявних серверів підприємства, використовуючи інформаційну систему PLM TeamCenter, з яких отримуються дані, що використовується в процесі створення складних технічних об'єктів.

Розроблені автором засоби інформаційної технології використані в процесі створення моделей літальних апаратів на ДП "Антонов". Реалізація інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів при проектуванні легкого транспортного літака та турбогвинтового регіонального вантажопасажирського літака на прикладі моделей вантажного люку забезпечує близько 22% економії матеріальних витрат. Основні положення та висновки дисертаційної роботи використані в навчальному процесі КПІ ім. Сікорського і Національного університету кораблебудування імені адмірала Макарова. Є відповідні акти впровадження.

Особистий внесок здобувача

Наукові положення, висновки і рекомендації, що виносяться на захист, отримано здобувачем самостійно. Із наукових праць, опублікованих у співавторстві, у дисертації використані лише ті ідеї, положення та матеріали, які є результатом власних досліджень автора. У роботах, написаних у співавторстві, автору належать наступні результати: в [79] проаналізовані проблеми взаємодії моделей та PLM; в [42] розкрито математичне забезпечення процесу побудови моделей; в [21, 22] формалізовано проблематику розробки інформаційних технологій для процесу побудови моделей; в [80] описано розподілену модель

клієнт/сервер для реалізації ІС PLM; в [24, 121, 122] розроблено модифікацію методу знання-орієнтованого створення СТО для керування параметричною інформацією при створенні моделей; в [136] представлені інформаційні технології обміну даними при створенні моделей; в [37, 38, 100, 150, 153, 154, 195] описано метод знання-орієнтованого створення складних технічних об'єктів та його модифікації; в [152, 158, 176] розкрита розробка засобів моделювання складних технічних об'єктів на основі знання-орієнтованих технологій; в [196] показані принципи реалізації знання-орієнтованого керування на попередніх етапах створення СТО.

Для виконання окремих завдань роботи здобувач пройшов стажування в Інституті інформатики Словацької академії наук (Братислава, Словаччина) та Корейському політехнічному університеті (Сеул, Республіка Корея) за підтримки конкурсних стипендій.

Апробація результатів дисертації

Основні результати досліджень, які включені в дисертацію, доповідалися та обговорювалися на: II International Conference «Methods and Systems of Navigation and Motion Control» (Kyiv, 2012); II International Conference «Actual problems of UAV development» (Kyiv, 2013); V Workshop on Intelligent and Knowledge Oriented Technologies (Bratislava, Slovakia, 2010); XIX Международной научной конференции студентов, аспирантов и молодых ученых «ЛОМОНОСОВ-2012» (Москва, 2012 р.); XVIII Міжнародній конференції з автоматичного управління “Автоматика/Automatics – 2011” (Львів, 2011 р.); II International Scientific Conference of Students and Young Scientists "Theoretical and Applied Aspects of Cybernetics" (Kyiv, 2012); XX-th International Conference «Knowledge-Dialogue-Solution» (Kyiv, 2014); IX, X та XI Міжнародних науково-технічних конференціях “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки” (Київ, 2013, 2015, 2017 рр.); II наук. конференції магістрів та аспірантів, присвяченій 20-річчю факультету прикладної математики (Київ, 2010 р.); Міжнародній науково-практичній конференції "Інформаційні технології та комп'ютерна інженерія" (Вінниця, 2010 р.); XII Міжнародній науково-практичній

конференції молодих учених і студентів «Політ. Сучасні проблеми науки» (Київ, 2012 р.); VII International Conference Students and young scientists «Intelligence. Integration. Reliability» (Kyiv, 2014) та інших, що приведені в дисертаційній роботі; наукових семінарах Інституту інформатики Словацької Академії наук (Братислава, Словаччина, 2010, 2011, 2016 рр.); відділенні перспективних розробок ДП «Антонов»; кафедрі автоматизації експериментальних досліджень КПІ ім. Сікорського; кафедрі приладів та систем керування літальними апаратами КПІ ім. Сікорського; науковому семінарі в авіаційній компанії САМАС (Шанхай, Китайська народна республіка, 2012 р.); науковому семінарі в Шеньянському авіаційному університеті (Шеньян, Китайська народна республіка, 2016 р.) та науковому семінарі наукової ради з напрямку «Інформаційні технології» (Київ, 2017 р.) Одержані наукові дослідження, зокрема, викладені у навчальному посібнику "Обчислювальні мережі: побудова та програмування": навч. посібник / Зінченко В.П., Зінченко С.В., Мірошниченко І.В., Конотоп Д.І., Резаї В. - К. : НТУУ «КПІ», 2011 . - 243 с., де описано розподілену модель клієнт/сервер для реалізації ІС PLM. Результати використовуються в навчальному процесі: при викладанні курсів на кафедрі автоматизації експериментальних досліджень КПІ ім. Сікорського: «Багаторівневі системи обробки інформації», «Обчислювальні мережі» та «Цифрові системи передачі даних».

Публікації

Результати дисертаційної роботи викладені в 24 наукових працях, з яких 1 [100] стаття у іноземному виданні; 1 [195] стаття входить до міжнародної наукометричної бази Index Copernicus; 7 [21, 37, 79, 121, 150, 154, 195] статей входять до Переліку наукових фахових видань, 14 [22, 24, 38, 42, 122, 136, 149, 152, 153, 155 – 158, 196] матеріалів і тез доповідей на наукових конференціях (2 з яких [38, 155] входять до міжнародної наукометричної бази SCOPUS) та 1 [80] навчальний посібник у співавторстві. 5 [24, 25, 136, 156, 157] статей належать особисто автору.

РОЗДІЛ 1

АНАЛІЗ ПРОБЛЕМАТИКИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ТА СУПРОВОДЖЕННЯ МОДЕЛЕЙ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ

Питання розробки проблемно-орієнтованих ІТ, зокрема створення СТО, впливають з задач підвищення якості проектування СТО. Вирішення цих питань вимагає всебічного аналізу проблематики розробки ІТ з метою вивчення сучасних (наявних) підходів та планування нових, також формування вимог до функцій та структури ІТ.

В розділі проведений аналіз наукових робіт, присвячених проблемі застосування ІТ при створенні СТО, проведений огляд наявних ІТ, що застосовуються при створенні різних етапів СТО, показано важливість побудови СТО, наведено математичний опис процесу побудови моделей СТО, поставлена задача розробки інформаційної технології створення інтегрованої моделі СТО.

1.1. Сутність наявного процесу створення складних технічних об'єктів

Створення СТО, які характеризуються великою кількістю елементів та різноманітних зв'язків (кораблі, автомобілі, літаки, ракети та інше), представляється розробкою наукоємного технічного об'єкта з розвиненою ієрархічною структурою, великим числом елементів та внутрішніх зв'язків. Сучасне створення СТО неможливе без застосування ІТ, які дають можливість системного підходу та оптимізації прийняття рішень з розробки та забезпечують високі показники ефективності СТО.

Процес створення СТО охоплює: маркетингові дослідження, проектування, технологічну підготовку виробництва, експлуатацію та утилізацію. Проектування СТО представляє собою багатоетапний процес (додаток А), основними класичними стадіями якого є [1 – 7]: етап створення вимог до СТО (зовнішнє проектування), результатом якого є формулювання технічних вимог (ТВ) та

надалі розробка технічної пропозиції (ТП); ескізне, технічне та робоче проектування (внутрішнє проектування) [8 – 10]. Етапи розробки ТП та ескізного проектування – найбільш відповідальні етапи створення СТО. На даних етапах при затраті до 20 % часу та до 10 % коштів приймається більш ніж 75 % основних технічних та організаційних рішень по проекту. [5] Тому, щоб мінімізувати повні витрати проекту та ризики розвитку, найбільша увага приділяється етапам розробки ТП та ескізного проектування.

Кожному етапу створення СТО відповідають свої моделі, що відрізняються ступенем деталізації. [3] Модель і моделювання – універсальні поняття, атрибути одного з найбільш потужних методів пізнання системи, процесу, явища в будь-якій професійній сфері. [11, 12] Моделювання – це представлення характеристик однієї фізичної або абстрактної системи за допомогою іншої системи. [13] Модель M_i містить: об'єкт O , суб'єкт (не обов'язковий) S , завдання T , ресурси R , середовище моделювання E :

$$M_i = \langle O, S, T, R, E \rangle.$$

На рис. 1.1 представлені основні моделі СТО, які отримані з різних складових IT CALS та PLM-рішень та відображають різні сторони його структури та функціонування [1] у вигляді діаграми класів відношення агрегації уніфікованої мови моделювання (unified modeling language – UML), де M_i – моделі: геометрична; вагова; аеродинамічна; компоновання та центрування (К/Ц); міцності; силової установки; технологічності; економічна відповідно. [13] Геометрична модель (ГМ) СТО – це базова інформаційна форма представлення СТО та його структури у сучасній концепції керування ЖЦ СТО, що визначає функціональні властивості СТО. Геометричне моделювання переплітається з обчислювальною математикою, диференціальною геометрією та алгоритмами роботи зі структурами даних. [14 – 16] Процес моделювання СТО в цілому, так і окремих його агрегатів, представляє собою визначену послідовність етапів та дій. [17] Відповідність класичних етапів проектування СТО та процесу моделювання СТО представлено на рис. 1.2. Ітераційна процедура моделювання СТО представлена на рис. 1.3. [17 – 23]

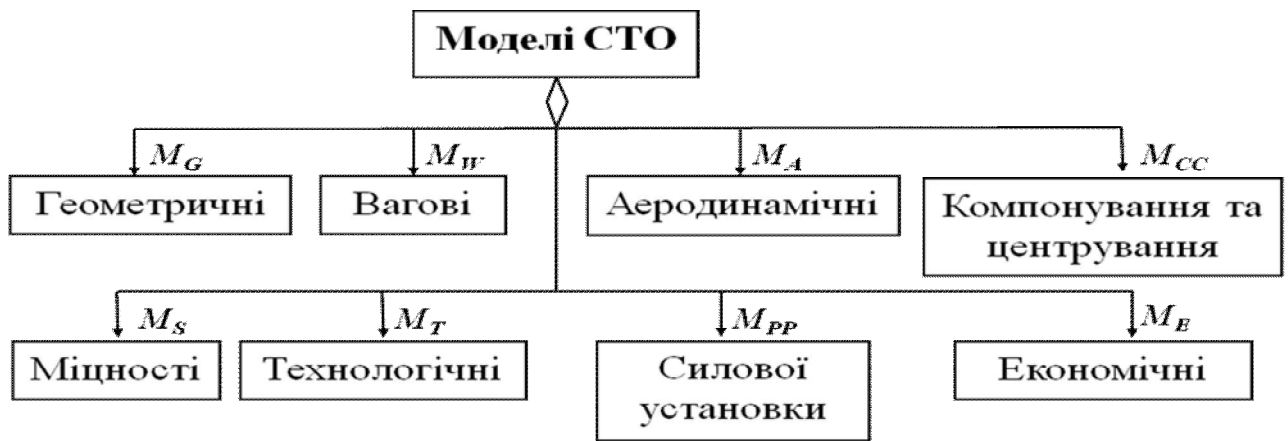


Рисунок 1.1 – Основні моделі складного технічного об'єкта



Рисунок 1.2 – Відповідність етапів проектування моделюванню СТО

На рис 1.3 показано: A0 та A1 – ТВ та ТП на розробку проекту СТО відповідно (зовнішнє проектування СТО); M1, M2, M3 – моделі СТО: базова контрольна структура (БКС), що містить модель майстер-геометрії (ММГ), модель розподілу об'єктів (МРО) та модель повного визначення (МПВ) СТО; C0, C1, C2, C3 – критерії для відповідного етапу проектування; R1, R2, R3 – методи експериментальних та чисельних досліджень розробки відповідного етапу проектування; GM – кінцева модель СТО.

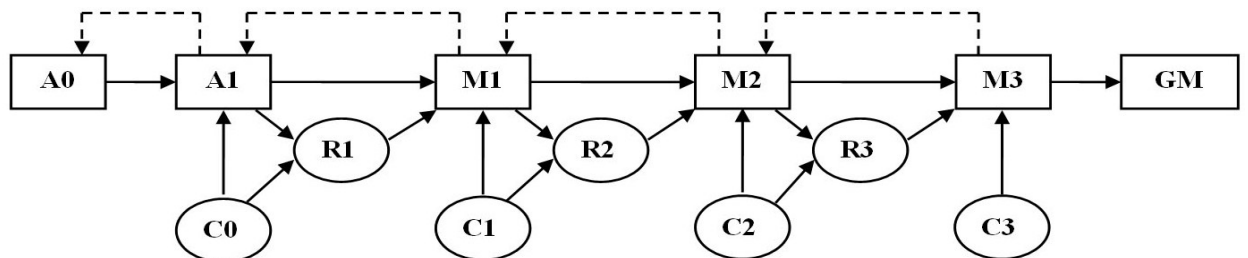


Рисунок 1.3 – Ітераційна процедура процесу моделювання СТО

Базова контрольна структура (БКС) є сукупністю вихідних даних, які є основою для розробки моделей СТО на будь-якій стадії проектування і

представляє собою ієрархічну структуру в ІС PLM [24, 25] та призначена для забезпечення спадковості вихідних даних на усіх етапах проектування. Базова контрольна структура – це математична модель об'єкта проектування з параметрами, при зміні яких відбувається зміна конфігурації моделей. На рис. 1.4 представлено склад БКС та у табл. 1.1 описані його складові.



Рисунок 1.4 – Склад базової контрольної структури

Таблиця 1.1 Опис складових базової контрольної структури

Складові БКС	Зміст складової	Позначення
Технічні рішення	Дані ТВ, ТП	ТР
Модель майстер-геометрії	Обрис моделі СТО	ММГ
Кінематична схема	Параметри моделі рухомих частин СТО	КС
Схема конструкції	Параметри конструкції СТО	СК
Схема розміщення систем та обладнання	Параметри моделі розміщення систем та обладнання	СРСО

Модель майстер-геометрії (ММГ, М1) представляє собою формування обрису СТО та містить у собі розробку моделей СТО на підставі ТВ та ТП. [26, 27] У додатку А представлено приклад ММГ.

Моделі розподілу об'єктів та повного визначення є складовими процесу внутрішнього проектування СТО.

Модель розподілу об'єктів (МРО, М2) СТО призначена для вирішення завдань попереднього проектування на рівні опрацювання СТО в цілому. Загальне число компонентів для МРО у дереві проекту СТО не перевищує 10000 одиниць. [28] Компоненти систем і обладнання формуються спрощеними параметричними моделями, що сполучаються з конструкцією СТО. [29] При компонованні МРО враховуються вимоги до розробки відповідних систем та їхнього спільного розташування. [17, 30]

Модель повного визначення СТО є опрацюванням РП. При створенні БКС, МРО і МПВ використовується технологія паралельного проектування PLM, котра забезпечує прямий і зворотний зв'язок поточних і попередніх етапів.

На рис. представлено 1.5 існуючий метод інтегрованого проектування та комп'ютерного моделювання СТО, котрий відображає сучасний процес створення СТО із використанням ІТ. [17, 31] Остаточним результатом проектування СТО є оформлення робочої документації (РД) і формування програм для верстатів з числовим програмним керуванням (ЧПК) для виробництва. [32, 33]

Створюючи моделі в ІС PLM та САх на стадії створення МРО чи МПВ, розробнику СТО необхідно використо-

увати інформацію з БКС та конструктивну обстановку. Для забезпечення інформацією, проектування здійснюється на базі технології зонування з використанням робочих структур (РС). [30] Розглянемо РС та зв'язки між компонентами (табл. 1.2).



Рисунок 1.5 – Метод інтегрованого проектування та комп'ютерного моделювання

Таблиця 1.2 Приклад робочої структури створення моделі

Складові робочої структури	Опис складових
Модель	Деталь або складальна одиниця моделі, яка розробляється
Базова контрольна структура	Використання параметрів з БКС тієї частини моделі, яка розробляється
Конструктивне оточення	Моделі, які знаходяться в області моделі, яка розробляється, отримані шляхом зонування

На підставі вихідних даних формуються схеми зонування проекту (додаток А). У додатку А представлено МРО СТО деревом проекту в ІС PLM.

1.2. Проблемні задачі процесу створення складних технічних об'єктів

Сучасне створення СТО, використовуючи технологію паралельного проектування в ІС PLM, дозволяє спростити і скоротити процес створення СТО, організувати та узгодити роботу великої кількості конструкторів при створенні проекту, і в цілому значно покращити характеристики СТО за рахунок автоматизації проектних робіт.

При цьому виникають суттєві проблеми, пов'язані з: швидкою переробкою інформації із моделей при внесенні змін від попередніх етапів проектування на подальші етапи проектування; передачею інформації з проектування на виробництво; зв'язком різноманітних моделей СТО; передачею даних між різними складовими ІТ CALS та PLM-рішень; організацією проектних даних.

Геометричні моделі описують відношення між параметрами СТО та характеристиками його форми та розмірів. [34, 35] Дані цієї моделі використовуються для вагових [36 – 38], аеродинамічних розрахунків [39, 40] та розрахунків на міцність [41], компонування СТО [42], графічного відображення результатів проектування [43], розробки технологічного оснащення та програм

для станків з ЧПК [44]. Зв'язок ГМ з моделями СТО описано на рис. 1.6, де показано, що ГМ пов'язані з ваговими моделями через масу ГМ; аеродинамічні – через аеродинамічні характеристики (АДХ) частин СТО; компоновання та центрування – через центр мас (ЦМ) ГМ СТО і т. д.

Ефективність проектування СТО обумовлена використанням системного підходу у вигляді таких його складових, як системотехніка і системний аналіз. В основі системотехнічного підходу лежать принципи максимуму ефективності системи; центрального домінування та декомпозиції системи.

Принцип максимуму ефективності свідчить, що найкращим варіантом проектованої системи є той, що володіє найбільшою ефективністю. Під критерієм ефективності системи розуміється співвідношення або різниця двох показників: показника цінності кінцевого результату, одержаного при експлуатації системи, і показника витрат на створення і експлуатацію системи.

Принцип центрального домінування дозволяє правильно вибрати цільові функції проектованих підсистем (локальні цільові функції), пов'язавши їх в загальносистемному плані з глобальною цільовою функцією.



Рисунок 1.6 – Приклад зв'язку моделей складного технічного об'єкта

Використання принципу декомпозиції дозволяє звести задачу проектування системи до безлічі завдань проектування її підсистем, пов'язаних у загальносистемному плані. Рівень складності завдання структурного синтезу стосовно до проектування моделей СТО вельми високий. Ефективне вирішення таких завдань можливе тільки при використанні IT CALS та PLM-рішень, де

приділено увагу формалізованій постановці задачі структурного синтезу. Основою такої формалізованої постановки є формулювання проблеми структурного синтезу як математичної задачі прийняття проектних рішень [45]:

$$ЗПП = \langle A, K, M, \Pi \rangle,$$

де $ЗПП$ – задача прийняття рішення; A – множина альтернатив проектного рішення; $K = (K_1, K_2, \dots, K_m)$ – множина критеріїв (вихідних параметрів), за якими оцінюються відповідність альтернативи поставленим цілям; M – модель, яка дозволяє для кожної альтернативи розрахувати вектор критеріїв; Π – вирішальне правило вибору альтернативи в багатокритеріальній ситуації.

Підсумовуючи огляд, існуючий метод інтегрованого проектування та комп'ютерного моделювання СТО, представимо ітераційною процедурою:

$$\begin{array}{ccccccc} \longrightarrow A_0 & \longrightarrow & M_1^i & \longrightarrow & \langle R_1 \rangle & \longrightarrow & M_2^i & \longrightarrow & \langle R_2 \rangle & \longrightarrow & M_3^i & \longrightarrow & \langle R_3 \rangle & \longrightarrow \\ & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \\ & & \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & & \underbrace{\hspace{1.5cm}} & & \\ & & (i+1) & & (i+1) & & (i+1) & & (i+1) & & (i+1) & & (i+1) & & \end{array},$$

де: A_0 – початкові дані (технічні рішення); M_i – модель на етапах розробки: БКС, M_1 , МРО, M_2 та МПВ, M_3 ; R_i – критерії для розробки кожного етапу розробки моделі; $(i+1)$ – зміни, уточнення кожного етапу розробки моделі. Моделі різних етапів створення СТО, для котрих основним є співвідношення "частина-ціле", представимо теоретико-множинною операцією об'єднання [46]:

$$M_i = \bigcup_{i=1}^N m_i, \quad (1.1)$$

де: m_i – компоненти моделей СТО етапу створення СТО, N – їхня кількість.

При багаторазовому вкладенні операцій об'єднання M_i зобразимо графом: $M_i = (A, U, P)$, де множини вершин A і ребер U , $U \cap M = \emptyset$ відбивають структуру об'єднання, а інцидентор P визначає упорядкування x, y, u : $x, y \in A, u \in U$. Поділ описів за ступенем деталізації відображуваних властивостей об'єкта лежить в основі блочно-ієрархічного підходу і призводить до появи ієрархічних рівнів в уявленнях про об'єкт. На рівні 0 A розглядається як система з n взаємно пов'язаних елементів і A_i на рівні 1. (рис. 1.7)

Кожен з елементів в описі рівня 1 представляє собою також СТО, який, в свою чергу, розглядаються як опис системи A_i на рівні 2. Елементами системи A_i є об'єкти $A_{i,j}$, $i, j = 1, 2, \dots, m_i$, де m_i – кількість елементів в описі системи A_i . Виділення елементів $A_{i,j}$ проходить по функціональній ознаці. Подібне розбиття триває аж до отримання на деякому рівні елементів, описи яких подальшому поділу не підлягають. [47] Принцип декомпозиції означає розбиття уявлень кожного рівня на ряд складових частин з можливістю роздільного проектування об'єктів A_i на рівні 1, об'єктів $A_{i,j}$ на рівні 2 і т.д.

При створенні СТО часто необхідно отримати з одних моделей інші шляхом формальних до визначень і додавань. У цьому разі розглянутий граф доцільно реконструювати, переорієнтувавши ребра по напрямку вкладення: деталь – система і отримаємо оргграф, що зростає, використовуючи принцип синтезу (агрегування) [48] елементів у складі моделей СТО (рис 1.8).

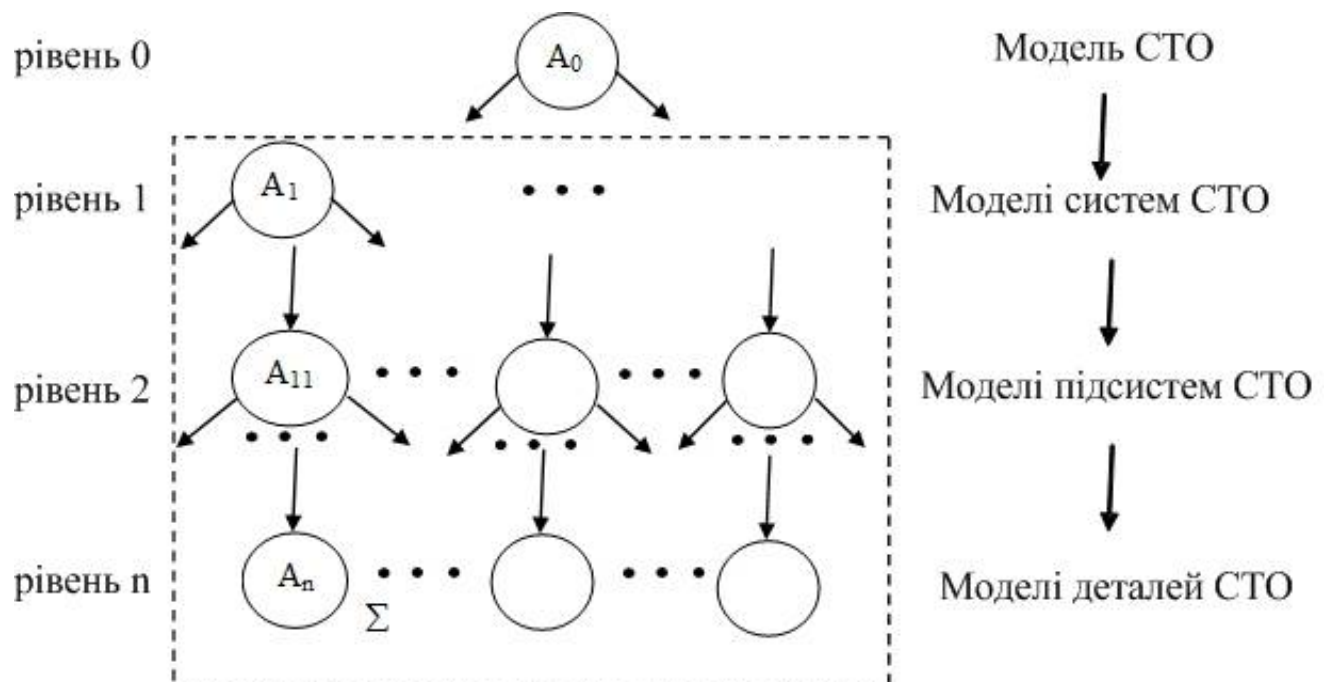


Рисунок 1.7 – Схема декомпозиції моделей складного технічного об'єкта

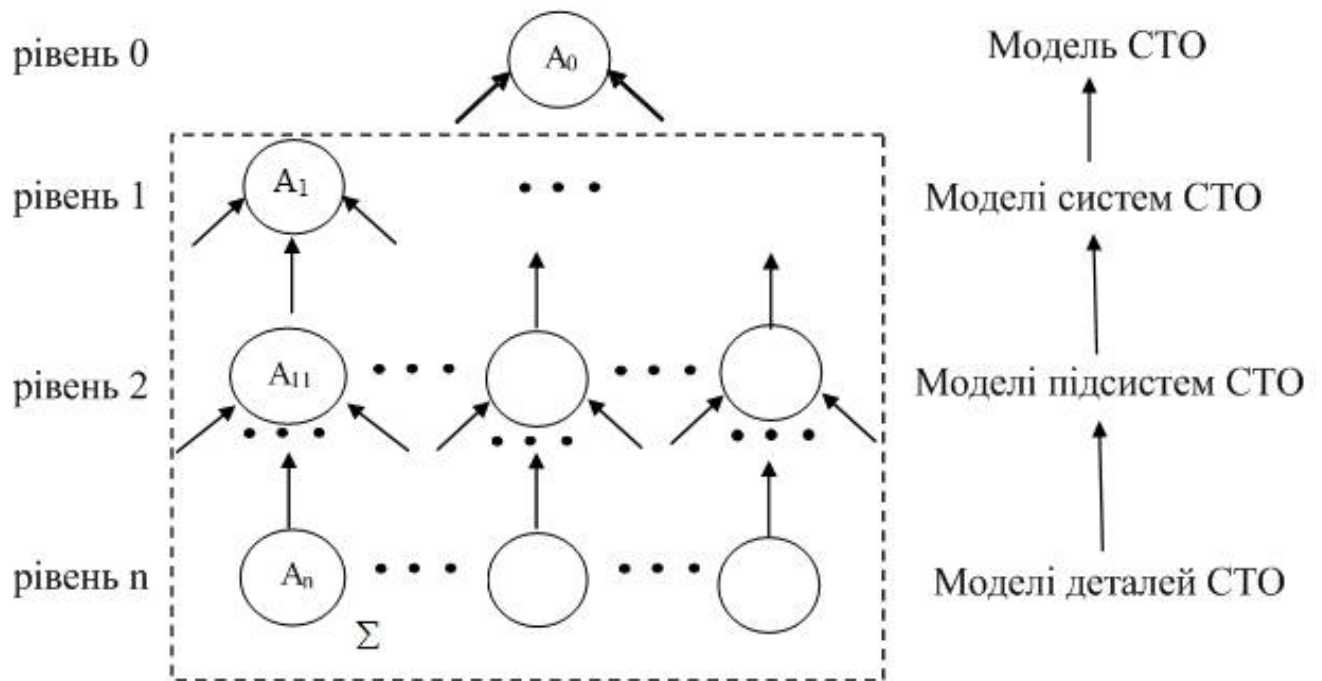


Рисунок 1.8 – Схема синтезу моделей складного технічного об'єкта

Розглянуті графи є основним концептуальним підходом до створення СТО, використовуючи IT CALS та PLM-рішення. [49]

Складність розкритих вище дерев $\sigma(T)$ створення моделей СТО визначається як сумарна складність їхніх вершин: $\sigma(T) = \sum_{M^{d_i} \in M} \sigma(M^{d_i})$. Визначимо складність

ярусу дерева створення моделей у вигляді: $\sigma(T)_i = \sum_{\lambda=1}^{\Lambda} b(M^{d_i})_{\lambda}$. Зростання дерева

між двома етапами створення моделей СТО визначається: $v_{i-1,i} = \frac{\sigma(T)_i}{\sigma(T)_{i-1}} \geq 1$.

Уздовж усього рівня проектування зростання дерева моделей характеризується:

$$V = \prod_{i=2}^r v_{i-1,i}$$

В даний час технологія керування даними про СТО на основі програмного забезпечення, що використовується на підприємствах, реалізується шляхом організації файлів проектування/виробництва, людей і керування процесами. PLM системи забезпечують середовище для спільної роботи за допомогою динамічного обміну і спільної візуалізації моделей за допомогою відкритого JT, VRML, або Java 3D. Проте всі ці стандарти, які є корисними для створення інтерактивної

взаємодії, певною мірою є обмеженими в ефективному поширенні моделей. Це відбувається тому, що доступ в мережу з обмеженою пропускнуою здатністю ефективної комунікації вимагає, в ідеалі, високу сприйнятливість до мінливого навколишнього середовища.

Побудова моделей, виконаних традиційними методами, переважно, не створюють несподіваних проблем, але вони є стомлюючими й, у такий спосіб, трудомісткими й дорогими. Більшість рутинних завдань при розробці СТО також є істотною причиною для того, щоб впроваджувати ЗОТ при розробці моделей СТО.

Підсумовуючи, представимо причини, які показують необхідність впровадження нових методів та засобів при побудові моделей СТО:

1. Моделі СТО містять безліч компонентів: деталей та складальних одиниць, створених за допомогою ІС PLM та САХ, багато міжкомпонентних параметричних виразів, посилань, обмежень і т.д., що вимагає детального опису процесу створення СТО, застосовуючи сучасні знання-орієнтовні технології.

2. Моделі на різних етапах створення СТО не пов'язані між собою. Тобто, якщо відбуваються зміни в компонентах моделей СТО на початкових етапах проектування, ці зміни не відображаються на робочому етапі створення СТО.

3. Різноманітні проектні моделі СТО фактично не пов'язані між собою та живуть своїм окремим життям в проекті з використанням ІС PLM та САХ, а багато даних взагалі знаходяться поза спільним проектом розробки СТО.

4. Процес створення СТО містить у собі безліч дисциплін та зв'язки між різними підрозділами. Для того, щоб скористатися одночасно технікою, інструментарієм проектування, бажано щоб проектування відбувалося паралельно з подальшим виготовленням та монтажем, що наразі є неймовірно складною, виснажливою та майже непов'язаною процедурою у зв'язку з слабкою автоматизацією виробництва, що унеможлиблює процес використання моделей, створених при проектуванні СТО.

5. Розробка моделей СТО відбувається, використовуючи різні ІТ CALS та PLM-рішення, що створює постійні складності конвертації даних з одних

програмних продуктів в інші та веде до часткової або повної втрати даних з моделей, її історії, топології та параметрів. Заради зниження часу виходу моделей СТО на ринок, а також для мінімізації помилок при відображенні моделей СТО, необхідно застосовувати нові підходи при передачі даних між різними ІТ CALS та PLM-рішеннями. Затримки при встановленні суперечливих питань при обміні даних, буде викликати багато переробок при створенні СТО і, таким чином план-графік проекту значно затримується.

1.3 Огляд наявних інформаційних технологій забезпечення життєвого циклу складних технічних об'єктів

Інформаційні технології – сукупність методів, виробничих процесів та програмно-технічних засобів, інтегрованих з метою збирання, опрацювання, зберігання, розповсюдження, показу і використання інформації в інтересах її користувачів [50]; орієнтовані на розв'язання певного класу завдань (рис. 1.9).

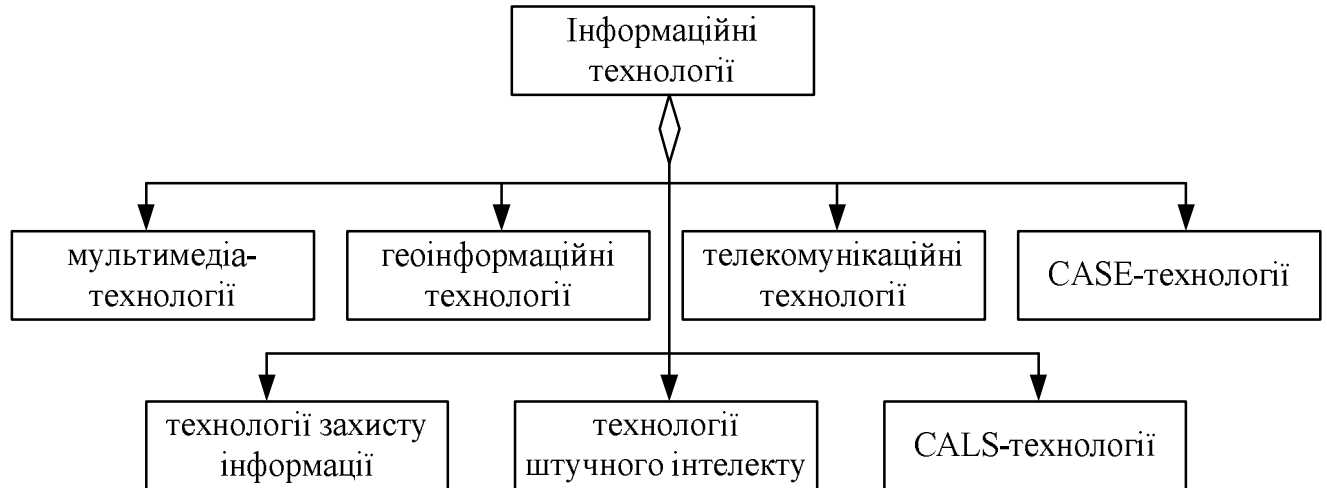


Рисунок 1.9 – Компоненти інформаційних технологій

Опису використання ІТ, яке створило «ефект найбільшої технологічної революції, що людство коли-небудь бачило» [51], у різних сферах людського життя приділено дуже багато уваги, особливо в роботах авторів з високорозвинених країн світу. [52] Зокрема, у "Journal of Information Technology Review" [53] друкуються матеріали, присвячені останнім новинкам, що

відбуваються в області застосування та створення ІТ. Потужне видавництво Wiley Periodicals публікує щорічний огляд інформаційних наук та ІТ. [54]

Роботи Глушкова [55] свого часу стали поштовхом для подальших розробок у сфері ІТ на теренах колишнього Радянського Союзу. В Україні видаються роботи, які присвячені як ІТ в загальному [56], різноманітним складовим ІТ: системному аналізу [57, 58], комп'ютерній графіці [59], веб-дизайну [60], ІС [61, 62] і т.д.

Інформаційні технології тісно пов'язані з ІС, що є для них основним середовищем, елементами якого є комп'ютери, комп'ютерні мережі (КМ), програмні продукти, бази даних (БД), люди, документація, програмні засоби зв'язку і т. д. Впровадження ІС підвищує ефективність діяльності підприємства за рахунок не тільки обробки й зберігання інформації, автоматизації рутинних робіт, але й принципово нових методів керування. [63] Інформаційні системи можна класифікувати, виходячи з цілого ряду різних ознак (рис. 1.10). [62, 64, 65] Автоматизовані ІС – системи, що реалізують ІТ виконання встановлених функцій за допомогою персоналу і комплексу засобів автоматизації. [62, 65]

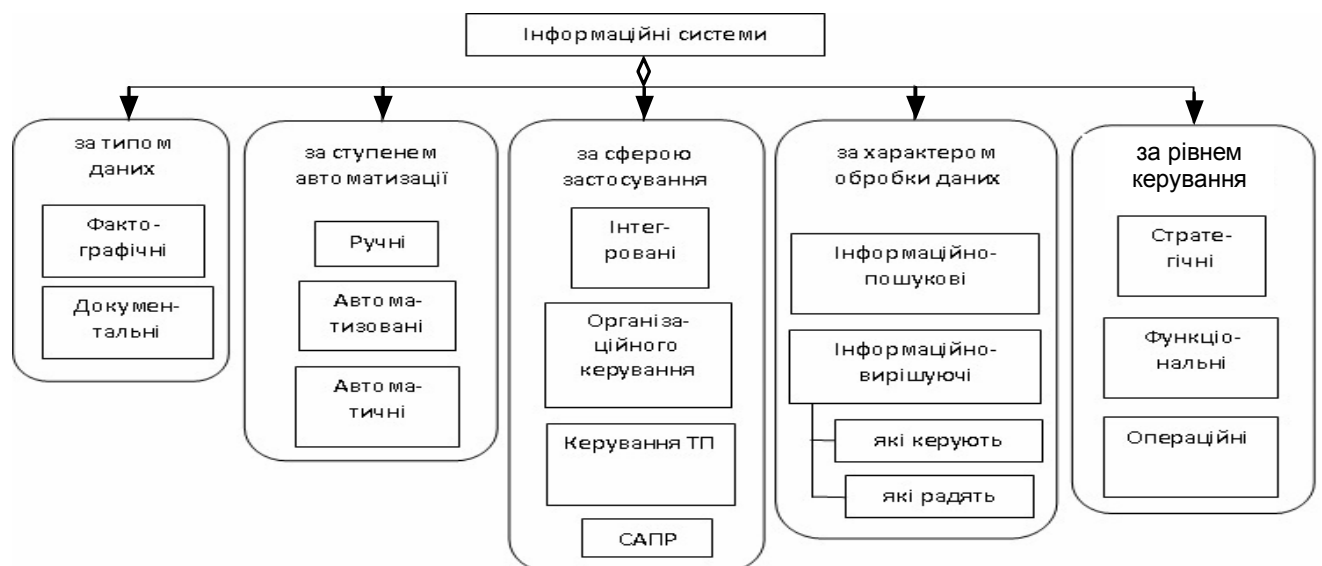


Рисунок 1.10 – Класифікація інформаційних систем

За словами Найджела Пікфорда, директора із досліджень ринку компанії SITA, лідера в області ІТ-послуг, роль ІТ при створенні СТО, зокрема, в авіації,

буде рости, оскільки до 2030 року очікується збільшення пасажиропотоку вдвічі. [66] Складні технічні об'єкти: літальні апарати (ЛА), судна, автомобілі, устаткування, пристрої, апарати та ін. складаються з ряду взаємодіючих технічних об'єктів, які функціонують на основі близьких процесів. Технічний об'єкт – будь-який виріб (деталь, вузол, підсистема, функціональна одиниця або система), який можна розглядати відокремлено. [67, 68]

З метою випуску нових продуктів кращої якості, низької вартості і за менший час сучасні підприємства використовують ІТ CALS та PLM-рішення [69, 70], призначені для уніфікації і стандартизації специфікацій промислової продукції СТО на всіх етапах його ЖЦ. [71] CALS є засобом, який інтегрує промислові автоматизовані ІС в єдину багатofункціональну систему. Життєвий цикл виробу включає ряд етапів від зародження ідеї продукту до його утилізації по закінченню терміну реалізації. [72] На рис. 1.11 представлений приклад ЖЦ виробу за Ібрагімом Зейдом [73]. У табл. 1.3 представлено ІС підприємства.

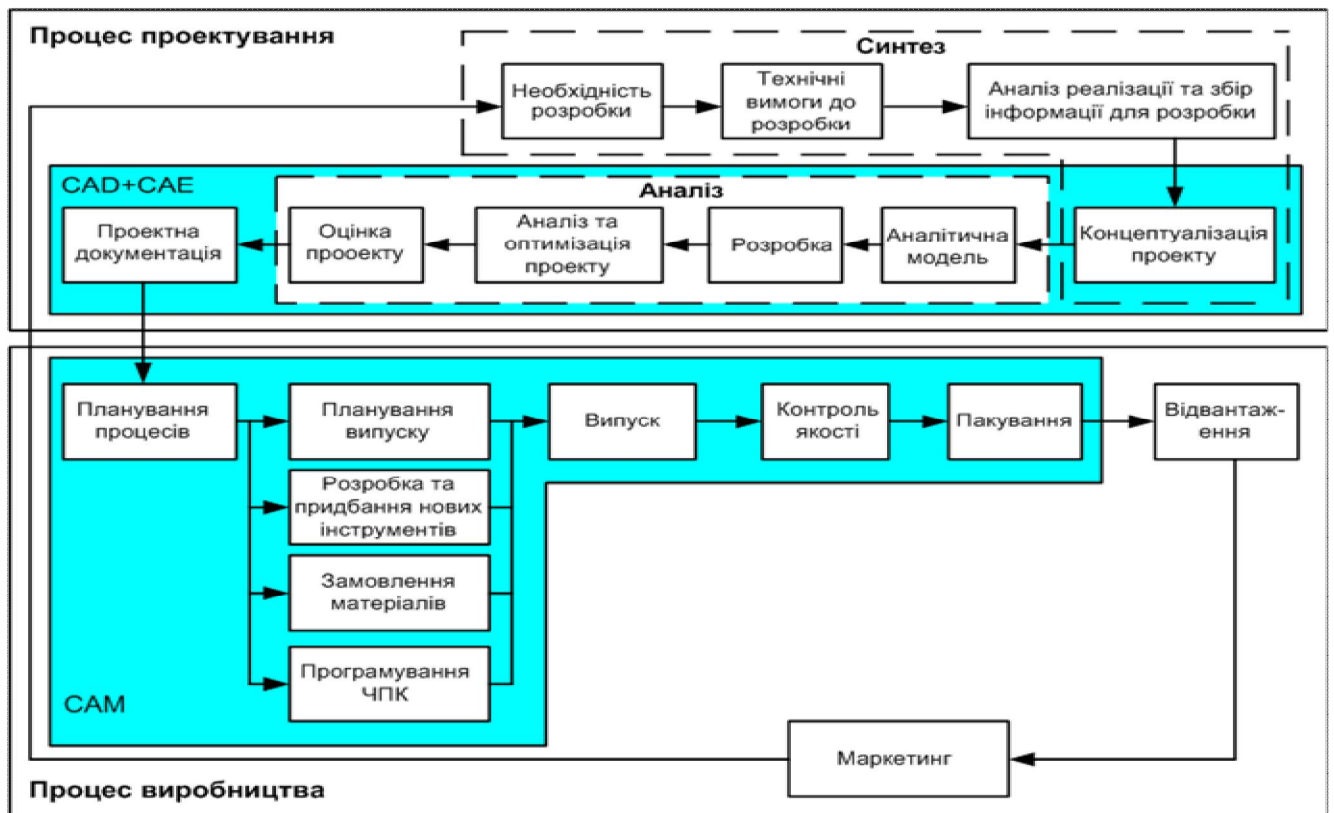


Рисунок 1.11 – Структура життєвого циклу виробу

Таблиця 1.3 Інформаційні системи підприємства

Інформаційні системи підприємства			
IC CRM (Customer Relationship Management – керування взаємовідносинами із замовниками)	IC SCM (Supply Chain Management – керування ланцюгами поставок)	IC ERP (Enterprise Resource Planning – планування ресурсами підприємства)	IC PLM (Product Lifecycle Management – керування життєвим циклом виробу)

Основними складовими IT CALS та PLM-рішень є IC, які використовуються на усіх етапах створення СТО: CAD (Computer Aided Design – комп'ютерне проектування); CAM (Computer Aided Manufacturing – комп'ютерне виготовлення); CAE (Computer Aided Engineering – комп'ютерний інженерний аналіз) [74]. В подальшому будемо використовувати термін "CAx" (Computer-Aided systems), розуміючи під ним IC комп'ютерного проектування, аналізу та виготовлення. [75] Здатність IC CAx працювати з файлами дозволяє систематизувати зберігання та забезпечити зручність пошуку документів. Для цього використовується IC PLM [76], яка виконує наступні функції: керування структурою та складом виробу; керування процесами та потоками робіт; керування збереженням даних та документів; механізму контролю доступу, авторизації; автоматичне генерування звітів, вибірок і т.д. [77, 78]

Інформаційні системи PLM доповнюють IC PDM функціональною «прив'язкою» усіх бізнес-процесів до структури виробу з виконання функціональності СТО на усіх етапах підтримки його ЖЦ, тому як інформаційне середовище виконання проектних робіт має підтримувати можливість зберігання інформації про всі зміни на підприємстві-виробнику та на суміжних підприємствах – учасниках процесів створення СТО. За допомогою IC PLM здійснюється відстеження великих масивів даних та інженерно-технічної інформації, необхідних на етапах проектування та виробництва, а також підтримка експлуатації, супроводу й утилізації технічних виробів.

Координація роботи підприємств-партнерів здійснюється за допомогою ІС СРС (Collaborative Product Commerce – керування даними в інтегрованому інформаційному середовищі).

Виробництво виробів також автоматизується. Для цього використовуються: ІС ERP, що забезпечує виробництво якісного виробу відповідно до вимог замовника при контролі за строками і витратами; та ІС MRP-2 (Manufacturing Resource Planning – планування виробничих ресурсів).

ІС MES (Manufacturing Execution Systems – виробнича виконавча система) призначена для вирішення оперативних завдань управління виробництвом.

Місце ІС PLM у загальному виробничому ланцюжку представлена схемою взаємин ІС САх і ERP на рис. 1.12. ІС PLM відіграють роль сполучної ланки між етапом інженерно-конструкторської підготовки нового СТО та ІС ERP.



Рисунок 1.12 – Взаємозв’язок систем автоматизації виробничих процесів

Відпрацьована технологія за допомогою Web-браузера фактично стала сьогодні корпоративним стандартом для доступу до інформації та реалізує принцип, коли регламентований доступ до додатку має бути приховано від користувача за звичним загальномережевим інтерфейсом, а інформаційний об’єкт довільної складності для додатку має керуватися одним й тим самим механізмом доступу.

Використання Web-браузера дозволяє кардинально удосконалити виробничі процеси та дає такі можливості: створення єдиного електронного архіву підприємства; керування конструкторським та технологічним документообігом; використання існуючих розробок в нових виробках; швидке конфігурування виробів за вимогами замовників; взаємодія спеціалістів підприємства; взаємодія з поставщиками та замовниками у реальному часі; захист інженерних даних та ноу-хау; керування робочими процесами ЖЦ

виробу; поставка актуальних даних в ІС ERP, CRM, SCM; оперативний випуск технічної документації.

Модульні, інтегровані рішення забезпечують повний набір інструментів для підтримки розробки виробів, керування інформацією і процесами, зберігання і керування технічною документацією; інтеграцію з різними ІС САХ та MES/ERP і т. д. (рис. 1.13)

З усіх методів інженерного аналізу найбільш широко використовується метод кінцевих елементів. [79] З його допомогою розраховуються напруги, деформації, теплообмін, розподіл магнітного поля, потоки рідин і інші завдання.

В ІС PLM реалізується розподілена модель клієнт/сервер і на основі стека протоколів TCP/IP підтримуються гетерогенні платформи, як на стороні клієнта, так і на стороні сервера. [80]; використовується операційні системи (ОС) Windows та UNIX від різних постачальників.

Для роботи з БД більшість ІС PLM реалізують так званий "нейтральний SQL-рівень" між кодом додатка та системами керування БД (СКБД). Потім розробляється спеціальний інтерфейс, що адаптує нейтральний SQL до певної системи. Основні ІС PLM базуються сьогодні на реляційних СКБД.

На сьогоднішній день існує багато постачальників систем і програмних пакетів для підтримки всіх етапів проектування або виробництва СТО, але лише компанії Autodesk, Dassault Systems, PTC, Siemens PLM Software є загальновизнаними найпотужнішими світовими виробниками. [81]

Згідно аналізу компанії Tech-Clarity [82], постачальниками ІС PLM є:

- "четвірка" PLM: Autodesk (Autodesk 360); Dassault Systemes (ENOVIA, 3DEXperience); PTC (Windchill); Siemens PLM Software (Teamcenter).

- постачальники – представники динамічного ринку, які мають велику кількість рішень в галузі PLM: Active Sensing; Aras; Arena Solutions; DEXMA; GrabCAD; Infor; Kenesto; Omnify Software; SofTech; Solid Edge SP (Siemens); SolidWorks (Dassault Systems); Synergis Software.

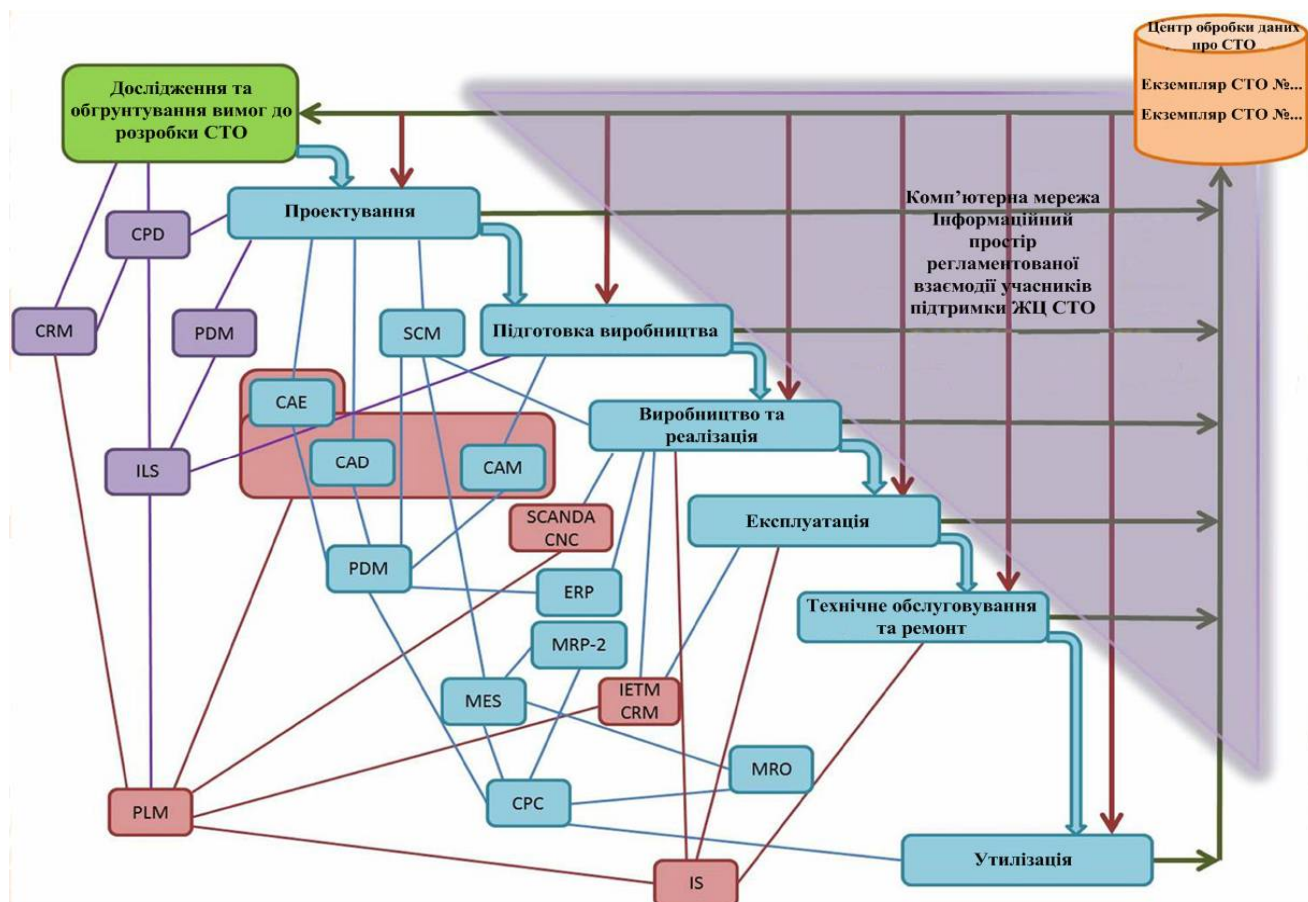


Рисунок 1.13 – Принципова схема взаємодії компонентів інформаційних систем керування повним життєвим циклом СТО

ІС PLM від компанії Autodesk [83] – Autodesk PLM 360 використовує хмарні технології і є інфраструктурою керування бізнес-процесами. Проте залишається проблема ізольованості керування даними.

ІС PLM від Dassault Systemes ENOVIA [84] (рис. 1.14) представляє собою рішення для великих та середніх підприємств, що використовують ІС САТІА. ENOVIA надає користувачам можливості спільної роботи над виробом в реальному часі.

Продукт Windchill компанії PTC [85] містить концепції Application Lifecycle Management та Service Lifecycle Management, що використовуються для механічного, електротехнічного та програмного проектування та намагається стерти межу між створенням та обслуговуванням виробу.

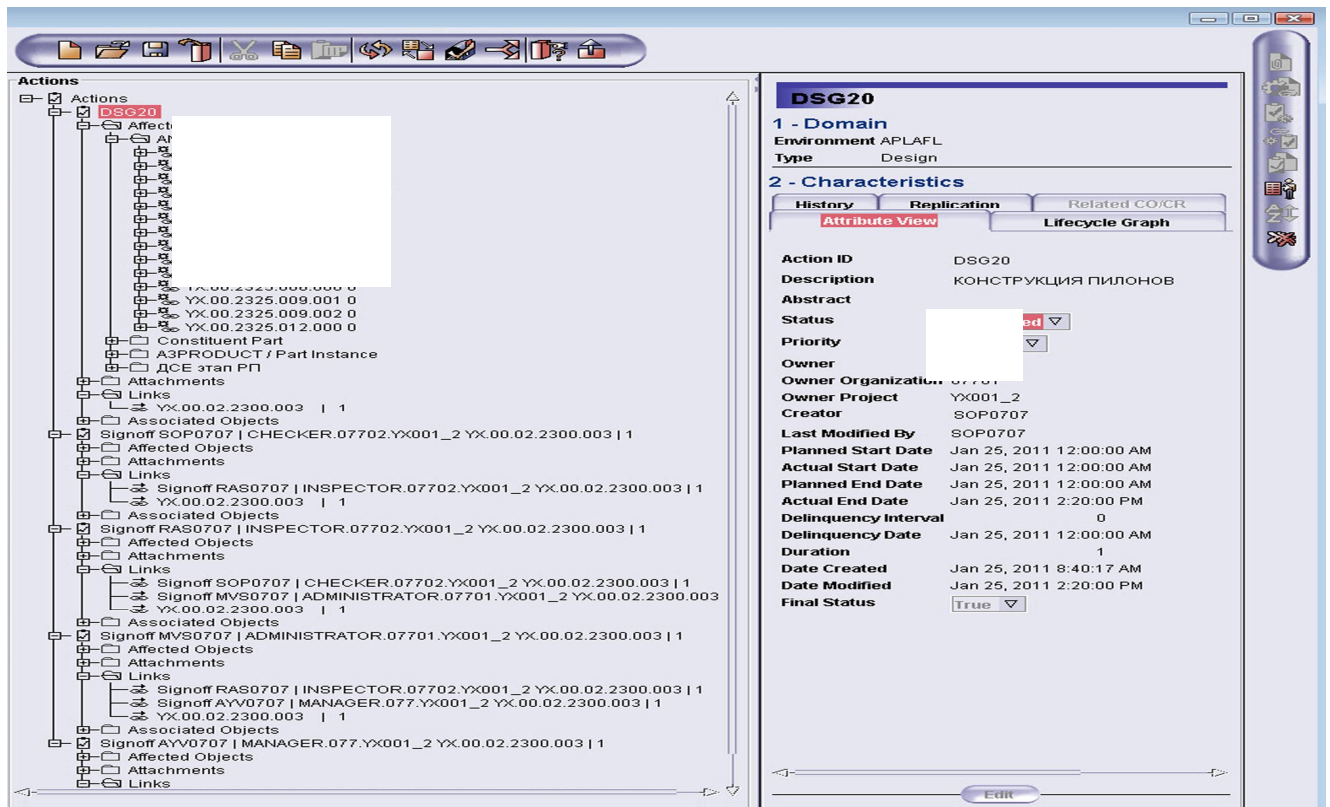


Рисунок 1.14 – Приклад ведення проекту СТО в IC PLM ENOVIA

IC PLM TeamCenter від Siemens PLM Software (рис. 1.15) має солідний набір PLM-рішень та концентрує багато сил для інтеграції PLM з виробничою інфраструктурою. І серед "четвірки" основних постачальників IC PLM наразі Siemens PLM є найбільш орієнтованою на клієнта компанією. TeamCenter керує більшою кількістю місць з IC CAx, ніж будь-яка інша PLM-система. [86] IC PLM TeamCenter володіє великими можливостями: керування конфігурацією виробу; можливість роботи з різними зовнішніми додатками, в тому числі різними IC CAx (так звана мульти-CAx); доступ користувача до функцій IC PLM при роботі в IC CAx (режим інтеграції в TeamCenter); підтримка паралельного проектування; створення моделі даних з урахуванням особливостей та вимог процесу розробки в конкретній галузі і т.д.

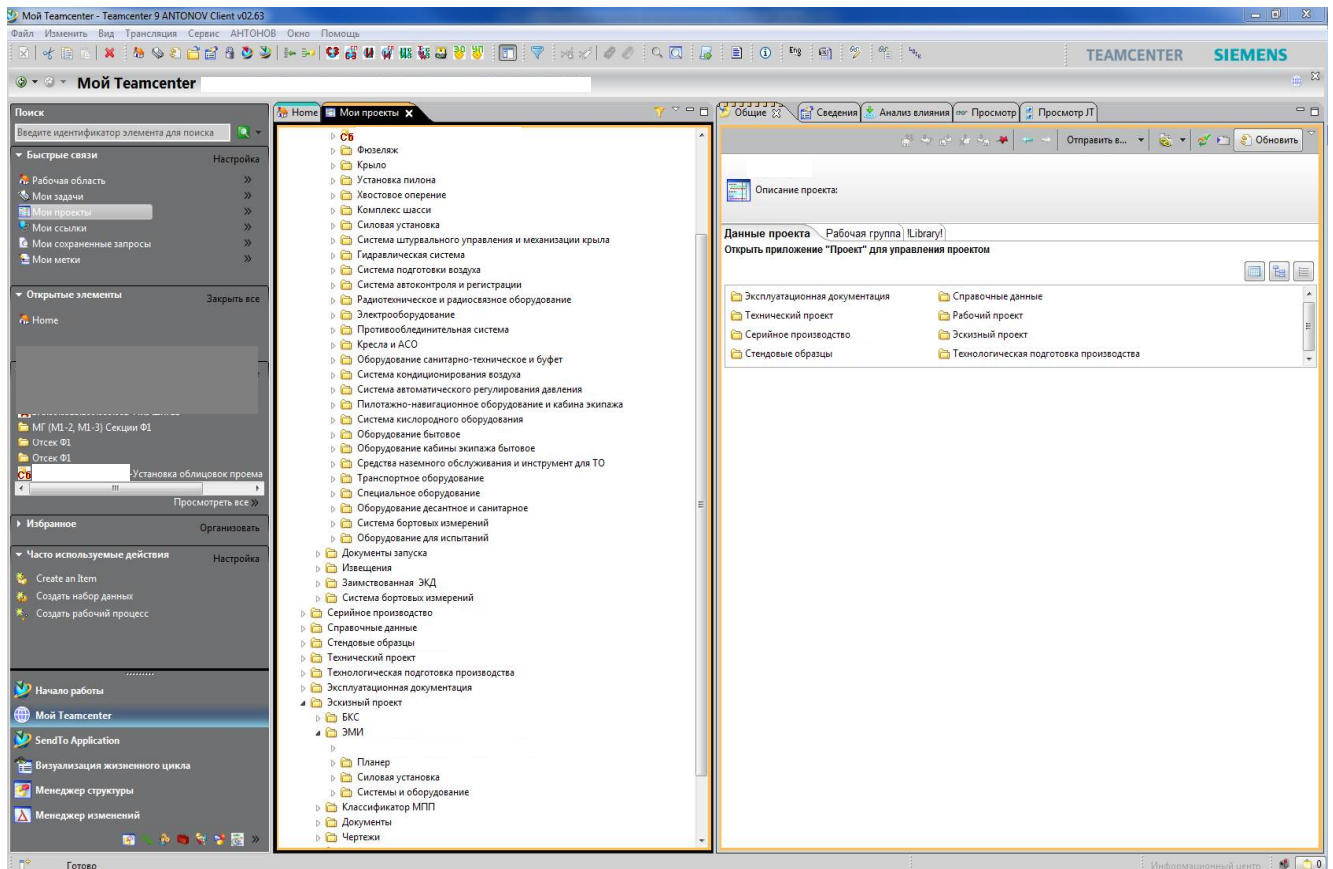


Рисунок 1.15 – Приклад ведення проекту СТО в IC PLM TeamCenter

На сьогоднішній день існує безліч постачальників IC CAx, але лише наступні програмні продукти є так званими "важкими" IC CAx: Creo або Pro/ENGINEER компанії PTC[83]; CATIA компанії Dassault Systemes [84]; NX компанії Siemens PLM Software [77, 86]. IC CAx, які розглядаються, містять: базу знань; зручний інтерфейс користувача; вміння генерувати на основі інформації нові знання і т. д. Розглянемо приклади використання знань в найпопулярніших IC CAx високого рівня.

Інформаційна система Pro/ENGINEER від компанії PTC містить елементи використання знань для ескізного проектування, має такі бібліотеки: типових двовимірних компоновань; типових тривимірних компоновок; типових ескізів, окремих деталей, типових конструктивних рішень [87].

Інформаційні системи CATIA, ENOVIA і DELMIA від компанії Dassault Systemes дозволяють повторно використовувати знання. Модуль "Знання про виріб" (Knowledgeware) групує функції, пов'язані зі знаннями та має наступні

підмодулі: консультант знань; експерт знань; шаблон знань про виріб; шаблон знань бізнесу-процесу. За допомогою електронних таблиць можна виділити спеціально обумовлені конструктором основні параметри двигуна. Керована методологія, що використовує геометрію, дозволяє швидко замінювати специфікації і багаторазово використовувати знання. [88]

Інформаційна система NX включає технологію DesignLogic, яка управляє виробом за допомогою закладених знань у формі функцій і формул, асоціативних розмірів і посилань. Модуль NX Knowledge Fusion є технологією застосування знань, де виробники можуть використовувати накопичені знання про виріб. У роботі [49] розглядається розробка внутрішніх та зовнішніх додатків для IC NX, виконаних програмуванням на мові C.

Всі вищеперераховані елементи використання знань, що містяться в різних складових IT CALS та PLM-рішень мають суттєві недоліки: знання, впроваджені в існуючі складові IT CALS та PLM-рішень, прив'язані безпосередньо до IC, в якій ведеться розробка СТО, що унеможлиблює використання універсальної бази знань в різних складових IT CALS та PLM-рішень. Зокрема, додатки, що написані на Java для IC NX не сприймає IC CATIA, котра зав'язана за програмування мовою Visual Basic і т. д.

1.4 Постановка задачі розробки інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів

Вищеперераховані проблемні задачі унеможливлюють процес використання єдиної моделі впродовж всього життєвого циклу СТО, використовуючи різноманітні IC PLM та CAx. Тобто, досі побудова моделей СТО, навіть використовуючи сучасні IT, є дуже складним процесом.

Вирішити ці проблемні задачі дозволить інформаційний процес формування узагальненої математичної моделі (УМ) СТО, котра дозволить вирішити задачу зв'язку моделей різних стадій життєвого циклу СТО. На базі моделі буде виконана розробка методу знання-орієнтованого створення СТО та інформаційної

технології створення та супроводження узагальненої моделі СТО. Зокрема, це використання сучасних парадигм інтелектуальних ІТ, як-от: бази знань та бази даних, які застосовуються при створенні СТО і виводять побудову моделей СТО на новий рівень.

Застосування знань дуже популярно для електронної комерції, інформаційної інтеграції, семантичної мережі та семантичних послуг. Нажаль, так само не можна сказати про процес створення СТО, зокрема про авіаційну, суднобудівну та автомобільну промисловості. Проект МОКА [89] розглядає деякі випадки використання знання в промисловості, але їх результати ніколи не видавалися. В [90] описується один сценарій, де використовується підхід застосування знань для поліпшення процесу тестування різних конфігурацій автомобілів. Але представлені знання не містять геометричної складової; операції проводяться тільки з декларативним знанням.

Згідно [89], відсоток часу виконання рутинних завдань при розробці СТО становить приблизно 80%. Створення проекту, який відповідає встановленим вимогам до нього, залежить значною мірою від творчого потенціалу безпосередньо розробника. Інакше кажучи, щоб мати успішний проект необхідно присвятити понад 20% часу розробки для творчих дій.

Представлення знань для моделювання СТО усе ще залишається невирішеним. У наступних розділах цієї роботи буде наведено останні дані з представлення знань, засноване на онтологічному підході за допомогою керування параметрами моделей СТО для зв'язку етапів розробки СТО та моделей розробки СТО між собою, що приносить багато переваг, як-от: зменшення вартості, повторне і якісне використання знань, швидша й більш налагоджена розробка СТО і т.д.

У зв'язку з вищеперерахованими проблемними задачами постає актуальне науково-практичне завдання: дослідити та розробити інформаційну технологію створення та супроводження узагальненої моделі СТО з метою зниження термінів створення СТО та зменшення їхньої вартості при забезпеченні відповідної якості

за рахунок налагодження зв'язку між моделями СТО та поєднання складових інформаційних технологій CALS та PLM-рішень.

На основі поставленого науково-практичного завдання розробляється концепція розробки інформаційної технології побудови моделей СТО, яка включає наступні складові: розробку узагальненої моделі СТО; розробку методу знання-орієнтованого створення СТО; розробку модифікацій методу (для: керування параметричною інформацією при побудові моделей СТО; обміну даними між складовими IT CALS та PLM-рішень) та засоби реалізації методу, що є базою для створення інформаційної технології.

Підсумовуючи, завданнями дослідження є:

1. Провести системний аналіз наявних технологій, методів та засобів створення моделей складних технічних об'єктів та інформаційних технологій CALS та PLM-рішень.

2. Сформулювати та дослідити узагальнену модель складних технічних об'єктів, яка дозволить налагодити інформаційний зв'язок між моделями створення складних технічних об'єктів.

3. Розробити метод знання-орієнтованого створення складних технічних об'єктів на базі узагальненої моделі, що дозволить забезпечити взаємозв'язок між етапами створення складних технічних об'єктів, а також поєднати складові інформаційних технологій CALS та PLM-рішень.

4. Створити модифікації методу знання-орієнтованого створення складних технічних об'єктів для: керування параметричною інформацією при створенні моделей складних технічних об'єктів; обміну даними між складовими інформаційних технологій CALS та PLM-рішень.

5. Розробити інформаційну технологію створення та супроводження узагальненої моделі складних технічних об'єктів та оцінити її ефективність.

1.5 Висновки

1. Описано існуючий процес створення моделей СТО. Проведений аналіз показав, що етап розробки ескізного проектування – найбільш відповідальний етап створення СТО. Цей етап характеризується великим потенціалом для модифікацій проекту СТО та високим ступенем інформації. Щоб мінімізувати повні витрати та ризики проекту, найбільше уваги приділяється етапам розробки попереднього та ескізного проектування.

2. Ефективність створення СТО значною мірою обумовлена використанням системного підходу у вигляді таких його складових, як системотехніка і системний аналіз. Представлена загальна схема декомпозиції та синтезу елементів в структурі моделей СТО.

3. Розглядаючи моделі СТО, було зроблено висновки, що саме моделі є основою створення СТО, без створення яких неможливо виконати основні розрахунки СТО.

4. Аналіз існуючих методів та засобів використання інформаційних технологій на різних етапах життєвого циклу СТО показує, що основними ІТ, які використовуються в процесі побудови моделей є використання технологій CALS та його компонентів. На основі критичного аналізу наявних методів та засобів застосування інформаційних технологій при побудові моделей на попередніх етапах проектування СТО були виявлені недоліки, які пов'язані з тим, що: модифікації в одних моделях СТО, отриманих з різних ІС, які мають нести навантаження на інші моделі, є величезними і схильними до помилок; взаємодія ІС технології CALS складна, що проявляється в проблемах конвертації даних з одних ІС в інші та веде до часткової або повної втрати історії створення моделей, їхньої топології та параметрів.

5. В процесі подальшого дослідження необхідно розробити метод та інструментальні засоби знання-орієнтованого створення СТО, що стануть теоретичною базою для розробки інформаційної технології створення та супроводження узагальненої моделі СТО.

РОЗДІЛ 2

РОЗРОБКА УЗАГАЛЬНЕНОЇ МОДЕЛІ ТА МЕТОДУ ЗНАННЯ-ОРІЄНТОВАНОГО СТВОРЕННЯ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ

У розділі описана концепція розробки інформаційної технології створення узагальненої моделі СТО; розглянуто розробку узагальненої моделі СТО, представлено проблеми оптимізації процесу моделювання складних технічних об'єктів, що представляє математичний апарат розробки інформаційної технології створення узагальненої моделі СТО.

В розділі наведено розробку методу знання-орієнтованого створення складних технічних об'єктів, що є базою для подальшої розробки інформаційної технології.

2.1 Концепція інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів

Задачею створення СТО є розробка схеми, структури та конструкції майбутнього СТО та складових його елементів, яка має забезпечити при визначених обмеженнях найбільш ефективно виконання поставлених цілей. Повністю автоматизувати процес створення моделей СТО неможливо, оскільки на ті рішення, які приймаються в даній області, впливає дуже багато факторів (технічних, економічних, соціальних). Головним завданням впровадження знань в процес створення СТО є максимально можливе спрощення для людини процесу автоматизації максимально можливої кількості задач. [91]

Велике значення у вирішенні проблеми представлення знань при розробці СТО мала розробка проекту МОКА. [89] Робота багатьох дослідників світу визначила життєвий цикл обробки знання для знання-орієнтованих технологій при розробці СТО та стала відправною точкою для розробки концепції знання-орієнтованої побудови частиною компаній-розробників ІС PLM та САх. У проекті МОКА використовувалася UML для представлення знань. Область використання

знань є дискусивною проблемою, деякі дослідники припускають, що будь-який об'єкт може бути представлений як область знань. Інші роботи суперечать цьому принципу [92] і пропонують будувати малі області, які легше підтримувати.

Але у будь-якому сенсі усі сучасні виробники усвідомлюють важливість використання знань при розробці СТО. У табл. 2.1 описано фактори, які впливають на мотивацію для застосування керування знаннями в 500 інтерв'юованих німецьких компаніях. [93]

Таблиця 2.1 Мотивація для керування знаннями в німецьких компаніях

Фактори мотивації	Значення фактору, %
Підвищення передачі знань для нових робітників	92
Покращення інтеграції знань в компанії	87
Захист компанії від втрати знань	82
Навчання робітників	73
Отримання недокументованих знань	72
Підвищення інновації	65

Знання-орієнтовані технології (ЗОТ) – технології, основними інструментами яких є знання та основною задачею яких є робота з цими знаннями. [94] В роботі впроваджуються знання-орієнтовані технології для поєднання таких основних складових процесу створення СТО: геометричного моделювання – це формування поверхневих і твердотільних моделей - субоб'єктів СТО; характеристики та обмеження СТО – опис методів і засобів, котрі використовуються при створенні СТО, наприклад, обмеження, що накладаються на конструкцію та обліковуються на всіх етапах створення СТО на основі ТВ та ТП; моделі СТО, отримані з різних складових IT CALS та PLM-рішень; складові IT CALS та PLM-рішень, які представляються інструментарієм підтримки життєвого циклу СТО та інтегрують промислові автоматизовані ІС в єдину багатофункціональну систему. На рис. 2.1 представлено місце ЗОТ при створенні узагальненої моделі СТО.



Рисунок 2.1 – Місце знання-орієнтованих технологій при створенні узагальненої моделі складного технічного об'єкта

На сучасному етапі розвитку ІС САх і PLM головний інтерес лежить у використанні знань при створенні СТО, що дозволило б подальше вдосконалення процесу проектування та виробництва СТО, і процес цей можна назвати: знання-орієнтоване створення моделей СТО, що включає використання прийнятного програмного забезпечення для придбання й багаторазового використання знань при створенні моделей СТО найбільш можливим комплексним способом. Використання ЗОТ при створенні моделей СТО пов'язане з багаторазовим використанням знань, які було отримано з попередніх розробок СТО.

Таким чином, при побудові моделей СТО, концептуально необхідно вирішити наступні задачі:

- провести системний аналіз ІТ створення моделей СТО;
- створити зв'язок моделей на різних етапах побудови СТО;
- пов'язати одні моделей з іншими моделями СТО, отримані з різних складових ІТ CALS та PLM-рішень;
- застосовувати ЗОТ при передачі даних між різними ІС САх та PLM.

Для ефективного вирішення перерахованих задач необхідно проаналізувати основні особливості предметної області побудови моделей СТО та розробити можливі методи та засоби проведення автоматизації даної області. Впровадження ЗОТ при створенні моделей СТО є результатом для прийому наступних потоків інформації:

- знання-орієнтоване створення моделей має керуватись адміністратором, який може змінювати необхідні дані в базах знань та даних;

- параметри можуть змінюватися за результатами виробництва;
- дані можуть коригуватися, якщо від керівництва компанією надходить змінені сформовані дані розробки СТО.

В розділі надалі розробляється узагальнена модель складних технічних об'єктів та представлено метод знання-орієнтованого створення СТО.

При створенні будь-якого СТО необхідний постійний контроль за розробкою ІТ, з яким працюють на підприємстві. Для цього розроблено модель оцінки необхідності внесення змін в існуючий метод залежно від зовнішніх факторів та внутрішніх можливостей. Одним із методів, що ґрунтуються на експертних оцінках, є SWOT-аналіз, який можна розглядати як комплексний метод попередньої оцінки та обґрунтування напрямів внесення змін до існуючої ІТ створення СТО на підприємстві-розробнику.

Методологія SWOT-аналізу передбачає спочатку виявлення переваг і недоліків, можливостей і загроз, потім встановлення зв'язків між ними, що в подальшому може бути використано для формулювання стратегії організації. [105] Наведемо варіанти факторів (табл. 2.2) в SWOT-таблиці, які впливають на оцінку доцільності та обсяги внесення змін в процес сучасного створення СТО. Створюється експертна комісія з $k=6$, де k – кількість експертів. Для визначення стратегії подальшого розвитку ІТ створення моделей СТО кожен експерт заповнює матрицю впливів можливостей на переваги системи. На перетині "Переваг" з "Можливостями" проставляється експертна оцінка їхнього взаємного впливу в балах від 0 до 5 (0 – відсутність впливу, ..., 5 – дуже сильний вплив). Для визначення стратегії подальших дій зі створення ефективної ІТ створення моделей СТО експерти заповнюють матрицю впливів можливостей (S) на переваги (O). На перетині S з O проставляється експертна оцінка їхнього взаємного впливу (e_{ij}) в балах від 0 до 5 (де "0" – "вплив відсутній"; "5" – "вплив дуже сильний").

Таблиця 2.2 Фактори, які впливають на оцінку доцільності та обсяги внесення змін в процес сучасного створення СТО

	Переваги (Strength, S)	Недоліки (Weakness, W)
Внутрішнє середовище	<ol style="list-style-type: none"> 1. Досвід роботи з різними ІС PLM та САХ 2. Ефективні моделі СТО 3. Зв'язок етапів створення моделей СТО 4. Конвертація даних між ІС 5. Нечіткі вимоги до засобів реалізації ІТ 	<ol style="list-style-type: none"> 1. Застаріле обладнання реалізації створення СТО 2. Складний процес відстеження зв'язку моделей різних етапів створення СТО 3. Невисокий рівень програмістського потенціалу
	Можливості (Opportunities, O)	Загрози (Threats, T)
Зовнішнє середовище	<ol style="list-style-type: none"> 1. Доопрацювання існуючих методів ІТ створення моделей СТО 2. Використання нових методів та засобів 3. Зниження строків створення СТО та матеріальних витрат 4. Вплив на суміжні галузі процесу створення СТО 5. Залучення розробників з вищих навчальних закладів 	<ol style="list-style-type: none"> 1. Складність впровадження модулів в існуючий процес створення моделей СТО 2. Розрив стосунків з суміжними партнерами або ВНЗ у сфері розробки СТО та постачання програмних продуктів 3. Вихід на ринок ІТ принципово нових підходів до процесу створення моделей СТО

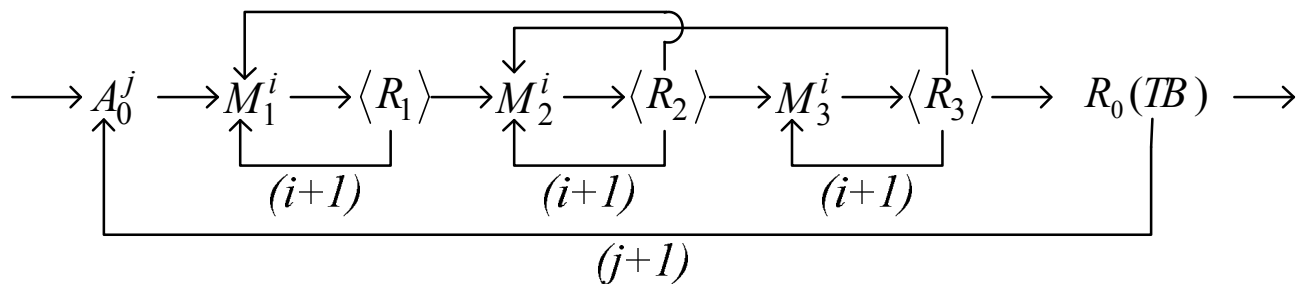
Підсумкова сума балів по стовпцях показує пріоритетність урахування того чи іншого чинника при виборі стратегії:

$$E_j = \max_j \sum_{i=1}^{n_s} e_{ij} (j = \overline{1, n_o}),$$

де e_{ij} – елемент матриці визначення оптимальної можливості, n_O – кількість можливостей, n_S – кількість переваг.

В результаті, ранжування можливостей набуває вигляду (ранг 1 присвоюється найкращому і т. д.): $O_1 < O_3 < O_2 < O_5 < O_4$, що показує найбільше значення при ефективному створенні СТО із використанням ІТ має доопрацювання існуючих ІТ створення моделей СТО для підвищення конкурентоспроможності та продовження життєвого циклу моделей СТО. На основі отриманої експертної оцінки надалі розробляється метод знання-орієнтованого створення СТО, який доповнює та розширює існуючий метод інтегрованого моделювання та комп'ютерного проектування та технологію паралельного проектування PLM з метою підвищення їхньої ефективності в процесі створення моделей СТО.

Представимо концептуально запропоновану ІТ створення та супроводження узагальненої моделі СТО ітераційною процедурою, що дозволяє пов'язати етапи побудови моделей відповідно критеріям ТВ за меншу кількість ітерацій:



де: $(j+1)$ – зміни відповідно критеріям ТВ ($R_0(TB)$).

Розробка ІТ створення та супроводження узагальненої моделі СТО на базі УМ та методу знання-орієнтованого створення СТО концептуально відбувається: в цілому, шляхом онтологічного опису процесу створення СТО; шляхом керування параметричною інформацією; та при обміні даними між різними складовими ІТ CALS та PLM-рішень.

2.2 Розробка узагальненої моделі складних технічних об'єктів

У зв'язку з вищеперерахованими проблемними задачами, постає первинна задача дослідження: після проведеного системного аналізу ІТ створення моделей СТО, дослідити та сформуванати узагальнену модель (УМ) СТО, яка дозволить вирішити задачу зв'язку моделей різних стадій життєвого циклу СТО; на базі моделі розробити метод та інформаційну технологію.

Після проведеного аналізу, опису та формалізації на рис. 2.2 представлено наявний процес створення моделей СТО в ІС PLM.

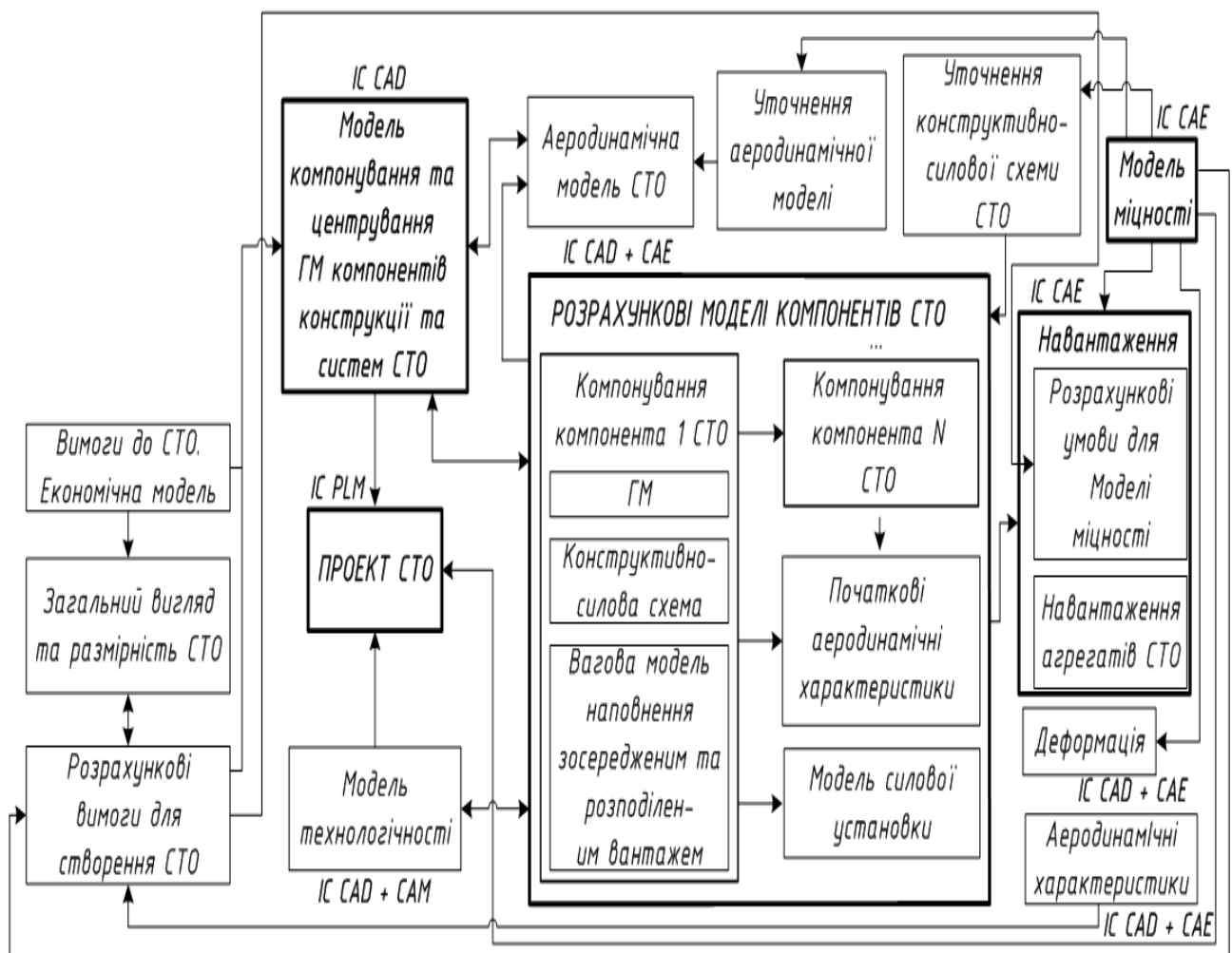


Рисунок 2.2 – Наявний процес створення моделей складних технічних об'єктів

Узагальнену модель СТО, що складається з набору основних моделей СТО, отриманих з різних ІС на різних етапах створення СТО представлено на рис. 2.3.



Рисунок 2.3 – Структурні елементи узагальненої моделі складних технічних об'єктів

Таким чином, узагальнену модель СТО представимо у вигляді сукупності:

$$M_U = \{M_G, M_W, M_A, M_{CC}, M_S, M_{PP}, M_T, M_E\},$$

де представлено наступні моделі: геометрична; вагова; аеродинамічна; компоновання та центрування; міцності; силової установки; технологічності; економічна відповідно.

Узагальнена модель СТО описується наступними типами інформаційних зв'язків (σ_{M_U}):

$$\sigma_{M_U} = \{\sigma_{M_U1}, \sigma_{M_U2}, \sigma_{M_U3}\},$$

де представлено зв'язки:

- 1). σ_{M_U1} – описані лінгвістичними змінними нечіткої логіки;
- 2). σ_{M_U2} – причинно-наслідкові (онтологічна схема створення узагальненої моделі СТО);
- 3). σ_{M_U3} – структурні (схема ІТ створення узагальненої моделі СТО).

Кожна модель компонента СТО описується власним набором параметрів, і представлена у вигляді сукупності: $m_i = f(P_i)$, де P_i – параметри моделей: $P_i = \bigcup_{i=1}^n p_i$.

Задача об'єднання моделей зводиться до вибору таких значень параметрів, при

котрих СТО буде відповідати ТВ та різних етапів створення. При цьому параметри моделей СТО мають відповідати обмеженням: $p_i^{\min} \leq p_i \leq p_i^{\max}$. Узагальнену модель компонента СТО можна представити наступним чином: $m_U = f^i(p_i, i = 1..n)$.

Узагальнена модель на основі використання теоретико-множинного підходу дозволяє налагодити зв'язок між моделями різних стадій життєвого циклу СТО та представляє вирішення k параметричних завдань $T_{i=1}^k$. Алгоритми розв'язку $T_{i=1}^k$ побудовано, як алгоритм мінімізації $P_{j=1}^n$ в умовах проектних обмежень:

$$\min(T) = \sum_{i=1}^k \min \left\{ \sum_{j=1}^n \bigcup_{ij} P_{ij} \right\}$$

На основі даного опису формується база правил, яка реалізується за допомогою бази знань (БЗ), бази даних (БД) та програмних модулів.

Позначимо $m_U = (m_1, m_2 \dots m_N)$ – множина УМ компонента СТО, кожна з яких включає обмеження (геометричні, технологічні і т. д.), $m_i \in M_i$.

Методи, що використовуються при створенні СТО для зв'язку різних моделей СТО та дані цих моделей можуть перебувати у конфлікті одна до одної. Для створення зв'язку між моделями СТО, щоб організувати й автоматизувати інструменти аналізу, надаючи проектувальникові здатність використовувати у своїх інтересах синергію між різними дисциплінами, використовують методи багатогалузевої оптимізації (БГО). [95] На реалізацію описаних методів витрачається велика кількість людино-годин і ресурсів. При цьому точність аналізу і час виконання забезпечення зв'язку між моделями може не задовольняти всіх спеціалістів та не відповідати заявленим характеристикам. Методи БГО застосовуються при створенні систем, де є взаємодія багатьох складових із різних галузей: $F(X_i, Z, R_j(X_i, Z))$, де: $i, j = 1..n, j \neq i, n$ – загальна кількість зв'язаних галузей, що представляє кожен галузь i ; де X_i – вектор локальної змінної; вектор R_j відповідає зв'язкам між галузями; Z відповідає вектору глобальної або об'єднуючої змінної; розглядаються параметри p_i для кожної галузі (рис. 2.4).

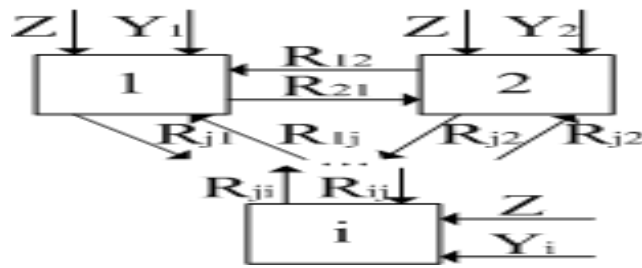


Рисунок 2.4 – Схема методу багатогалузевої оптимізації

Задача об'єднання параметрів СТО, отриманих з різних моделей в узагальненому вигляді зводиться до вибору таких значень параметрів та встановлення обмежень ΔX_i та ΔZ , при яких: $\min F(X_i, Z, R_j(X_i, Z))$ та є відповідність усім вимогам, що висуваються до СТО на різних етапах життєвого циклу при вирішенні задач проектування та виробництва СТО. Для цього, вихідні параметри мають відповідати обмеженням: $A_i \leq p_i(g, x, q) \leq B_i$, $i = \overline{1, n}$, що є деякою функцією геометричних, внутрішніх та зовнішніх параметрів; A_i та B_i – межі припустимих змін i -го вихідного параметру, що описується у вимогах.

Багатоекстремальність задачі передбачає пошук локальних екстремумів та їхнього перебору для пошуку глобального екстремуму цільової функції.

Наразі відсутній автоматизований зв'язок поєднання даних між етапами проектування та виробництва, навіть використовуючи ІС PLM. На рис. 2.5 представлено приклад узагальненої моделі СТО, за рахунок використання моделі технологічності і поєднання її з моделями СТО, отриманих з різних ІС.

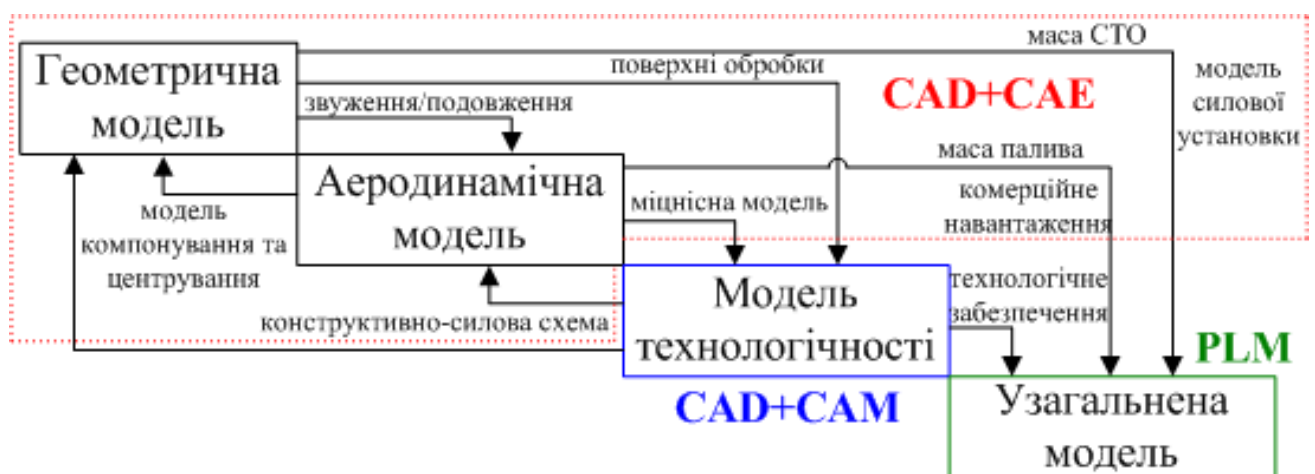


Рисунок 2.5 – Приклад узагальненої моделі складних технічних об'єктів

2.3 Математичне забезпечення процесу створення узагальненої моделі складних технічних об'єктів

Особливістю завдання розробки інформаційної технології створення узагальненої моделі СТО, яка використовується при оптимізації процесу створення моделей СТО як частинних компонентів СТО, так і СТО в цілому, є те, що число критеріїв оптимальності $n \gg 1$, і завдання можна вирішити, використовуючи методи аналізу, декомпозиції і синтезу. [96, 97] Тобто, процес реалізується як послідовне вирішення незалежних підзадач, для кожної з яких визначені окремі критерії якості і рівні декомпозиції. Для цього запропоновано розглядати процес створення СТО як сукупність субоб'єктів відповідно до їх призначень, що дозволить виділити і згрупувати окремі критерії і, в підсумку, розробити ІТ створення УМ як частинних субоб'єктів СТО, так і СТО в цілому. Показано, що проблема синтезу вирішена тільки для вузького кола завдань (синтез моделей складальних одиниць і т. д.)

Проблема оптимізації процесу створення моделей СТО формулюється як задача процесу прийняття рішення, який включає три етапи: «зовнішня» розробка (визначаються цілі та завдання, і формуються вимоги до її характеристик, що забезпечують досягнення цих цілей), «формування образу» і «внутрішня» розробка (реалізація основних параметрів моделей).

Розбиття процесу створення моделей СТО на етапи обумовлено тим, що безпосереднє визначення шуканого результату процесу на всій мислимій безлічі її варіантів практично неможливе. Це пов'язано з тим, що складові ІТ CALS та PLM-рішень, які застосовуються при створенні СТО, вимагають значних ресурсів на один варіант [98]. Отже, на етапі «формування образу» необхідно зменшити число альтернативних варіантів проекту з урахуванням вимог «зовнішньої» розробки, використовуючи метод неформальної декомпозиції [99].

Позначимо $m_U = (m_1, m_2 \dots m_N)$ – множина моделей компонента СТО, що розробляється, кожна модель включає визначені обмеження (геометричні,

центрувальні, вагові, функціональні, і т. д.) в залежності від типу параметра, $m_U \in M_U$. [100]

Для єдиного критерію ефективності $F_i(m_U)$ моделі $m_U \in M_U$ задача оптимального процесу створення узагальненої моделі СТО з урахуванням обмежень, накладених на етапі побудови БКС, полягає у визначенні вектора параметрів процесу створення СТО:

$$m_U^0 \in \text{Arg max}_{m_U \in M_U} F(m_U),$$

$$\text{де: } \text{Arg max}_{m_U \in M_U} F(m_U) = \{m_U \in M_U \mid F(m_U) = \max_{m'_U \in M_U} F(m'_U)\} \quad (2.1)$$

Кожен компонент моделі описується власним набором параметрів, і може бути представлений у вигляді сукупності: $m_i = f(P_i)$, де P_i – параметри моделей:

$$P_i = \bigcup_{i=1}^n p_i. \text{ Узагальнену модель компонента СТО можна представити наступним}$$

чином: $m_U = f^i(p_i, i = 1 \dots n)$.

Розробка УМ компонента СТО пов'язана з великою розмірністю N множини m_U , що потребує великих ресурсів для визначення значень $F(m_U)$. Тому для вирішення (2.1) необхідно використовувати метод декомпозиції.

Якщо $u_i(m_U)$ ($i = 1, 2, \dots, e$) є частинними критеріями якості процесу створення узагальненої моделі СТО, $F(m_U)$ монотонний (для $\forall m_U, m'_U \in M_U$ з $u_i(m'_U) \geq u_i(m''_U), i = 1, 2, \dots, e \Rightarrow F(m'_U) \geq F(m''_U)$). Позначимо $\Pi(U)$ – множина оптимальних за Парето векторів з U , $\Pi(M_U)$ – множина векторів $m_{VM} \in M_{VM}$ для яких $u(m_U) \in \Pi(U)$, тоді задача визначення вектору параметрів УМ компонента СТО набуває вигляду:

$$m_U^0 \in \text{Arg max}_{m_U \in \Pi(M_U)} F(M_U) \quad (2.2)$$

Отже, розв'язок (2.1) можна представити як декомпозицію: знаходження векторів $m_U \in \Pi(M_U)$ та розв'язання (2.2). При цьому етап "формування образу" не залежить від критерію F , що дозволяє істотно скоротити число варіантів узагальненої моделі СТО на етапі "зовнішньої" розробки. В результаті на цьому етапі формується узгоджене з можливостями "внутрішньої" розробки ТВ. Рішення

(2.2) розв'язує задачу оптимального процесу створення ІТ (2.1), де процедура "формування образу" складається з побудови паретовської множини $\Pi(u, M)$. Таке ж завдання має вирішуватися і на етапі "внутрішньої" розробки. Незважаючи на відносну "простоту" частинних критеріїв, завдання побудови множини $\Pi(u, M_U)$ є непротим через велику розмірність вектора $m_U = (m_1, m_2 \dots m_N)$ та складної структури множини M_U . Виходом з цієї ситуації є подальша декомпозиція і створення ієрархії задач "внутрішньої" розробки.

Відзначимо, що на практиці декомпозиція буде неформальною, тому що вибір вектора частинних критеріїв необхідно узгоджувати з можливостями як «внутрішньої», так і «зовнішньої» розробки узагальненої моделі СТО з використанням ІТ. Для правильного вибору вектора досліджуються паретовські множини розробки ІТ, які відповідають різним векторам частинних критеріїв (різні концепції). В [101] запропонований формальний підхід для перевірки виконання умов монотонності $F(m_U)$. На практиці умова монотонності випливає з самого сенсу частинних критеріїв, обраних розробником. Ця умова є умовою виконання декомпозиції, тобто, якщо критерій $F(m_U)$ задовольняє умові монотонності, тоді декомпозиція (2.1) можлива.

Насправді ефективність процесу створення СТО визначається не тільки вектором параметрів її розробки m_U , але і зовнішніми по відношенню до СТО умовами. З точки зору розробника УМ, ці умови можна описати як невизначені фактори, значення яких заздалегідь невідомі. Рішенням «зовнішньої» розробки власне процес створення СТО не закінчується, тому що з'являється проблема деталізації СТО, розробки підсистем і зв'язків між ними. Це етап «внутрішньої» розробки [102].

Припустимо, що узагальнена модель СТО з необхідним ступенем детальності описується вектором $m_U(0) \in R^{N_0}$ розмірністю N_0 , причому відомо, що $m_U(0) \in M_{U0} \subset R^{N_0}$. Це означає, що завдання конкретного значення $m_U(0) \in M_{U0}$ цілком визначає варіант узагальненої моделі СТО, тобто створюється узагальнена модель СТО з вектором параметрів $m_U(0)$. Множина M_{U0} , що реалізує СТО,

обмежена фізичними законами. Відзначимо, що $m(0)$ може забезпечувати далеко не «найдетальніший» опис узагальненої моделі СТО. Вибір опису СТО (вектора параметрів), впливає з двох вимог: повинен бути досить докладним, на деяких етапах для цього замість кінцевомірного простору R^{N_0} можуть вимагатися простори функціональні (наприклад, опис підсистем); повинен бути досить простим. Таким чином, при виборі вектора параметрів СТО потрібно знайти компромісне рішення між точністю опису об'єкта і його простотою. Якщо такий компроміс існує, то основним завданням є визначення $m_U(0)$ (вектор агрегованих параметрів нульового рівня), який можна виразити через інші параметри, відповідні більш детальному опису СТО.

Рішення завдання створення узагальненої моделі СТО для відомого векторного критерію ефективності $E_0(m_U) = (E_0^1(m_U), \dots, E_0^{n_0}(m_U))$ за єдиного критерію оптимальності F_0 зводиться до пошуку:

$$m_U^*(0) \in \text{Arg max}_{m_U(0) \in M_{U_0}} F_0(m_U(0)) \quad (2.3)$$

або до знаходження $\Pi(E_0, M_{U_0})$, (2.4)

де $\text{Arg max}_{m_U \in M_U} F_0(m_U) = \{m_U \in M_U \mid F(m_U) = \max_{m_U \in M_U} F_0(m_U)\}$; $\Pi(E_0, M_{U_0})$ – множина точок Парето з M_{U_0} за сукупності частинних критеріїв $(E_0^1(m_U), \dots, E_0^{n_0}(m_U))$.

Рішення (2.3) та (2.4) для ІТ неможливі внаслідок великої розмірності вектора N_0 та складності припустимої множини параметрів M_{U_0} , глобального критерію F_0 , векторного критерію E_0 . Вихід із цього становища – це побудова ієрархії процесу «внутрішньої» розробки.

Введемо агреговані параметри першого рівня: $m_U(1) \in M_{U_1}, M_{U_1} \in R^{N_1}, N_1 < N_0$, де $m_U(1) = f_1(m_U(0))$, $M_{U_1} = \{m_U(1) = f_1(m_U(0)) \mid m_U(0) \in M_{U_0}\} \equiv f_1(M_{U_0})$ – образ M_{U_0} при відображенні f_1 . Введемо критерії першого рівня $E_1(m_U(1)) = (E_1^1(m_U(1)), \dots, E_1^{n_1}(m_U(1)))$.

Оскільки $N_1 < N_0$, та розмірність вектора параметрів зменшилась, тоді $m_U(1)$ забезпечує більш цілісне, ніж $m_U(0)$, представлення про узагальнену модель СТО. Величини f_1 , E_1 мають бути визначеним чином узгоджені з E_0 , $m_U(0)$, M_{U_0} . Наприклад, процес створення компонента узагальненої моделі СТО описується вектором $m_U(0)$ значної розмірності. Вектор $m_U(1)$ може представляти собою вагові характеристики компонента узагальненої моделі СТО, еквівалентного $m_U(0)$. Критеріями E_0 та E_1 можуть бути значення, які враховують різний ступень деталізації описів $m_U(0)$ та $m_U(1)$.

Якщо й на цьому рівні задача побудови паретовської множини $\Pi(E_1, M_{U_1})$ варіантів компонента СТО, які відповідають набору частинних критеріїв $E_1(m_U(1))$ надто складна, тоді вводиться наступний рівень синтезу і так далі до декотрого s -го рівня $m_U(s) = f_s(m_U(s-1)) \in M_{U_s}$, $M_{U_s} \subset R^{N_s}$, $N_s < N_{s-1}$, $M_{U_s} = \{m_U(s) = f_s(m_U(s-1)) \mid m_U(s-1) \in M_{U_{s-1}}\} = f_s(M_{U_{s-1}})$, коли задача формування множини $\Pi(u_s, M_{U_{s1}})$ може бути вирішена за прийнятний час.

Така декомпозиція неоднозначна і реалізація навіть одного варіанта синтезу вимагає значних ресурсів. Тому, необхідно зменшувати число етапів синтезу, однак при цьому збільшується ймовірність помилки при переході від одного опису ІТ до іншого.

Після виконання всіх етапів синтезу, тобто введення величин m_{U_k} , M_{U_k} , E_k , $0 \leq k \leq s$, задачу «внутрішньої» розробки можна вирішити наступним чином. Знайдемо $\Pi(E_s, M_{U_s})$ та всі рішення рівнянь $f_s(m_U(s-1)) = m_U(s)$ з множини $M_{U_{s-1}}$, коли $m_U(s)$ пробігає множину $\Pi(E_s, M_{U_s})$.

Ці рішення визначають множину $\bar{M}_{U_{s-1}} = f_s^{-1}(\Pi(E_s, M_{U_s}))$, де $f_s^{-1}(\Pi)$ – повний прообраз множини Π при відображенні $f_s(\bullet)$. Далі вирішуємо задачу визначення множини $\Pi(E_{s-1}, M_{U_{s-1}})$ та $\bar{M}_{U_{s-2}} = f_{s-1}^{-1}(\Pi(E_{M_{U_{s-1}}}, \bar{M}_{M_{U_{s-1}}}))$. Продовжуючи цей процес, через s кроків прийдемо до задачі знаходження

$\Pi = \Pi(E_0, M_{U_0})$. Цей процес можна представити у вигляді рекурентного рівняння процесу створення узагальненої моделі СТО:

$$\Pi_k = \Pi(E_k, f_{k+1}^{-1}(\Pi_{k+1})), k = s-1, s-2, \dots, 1, 0, \Pi(E_s, M_{U_s}). \quad (2.5)$$

Рішенням задачі «внутрішньої» розробки буде множина Π_0 .

На кожному кроці рішення рекурентного рівняння відбувається відбір, що призводить до скорочення числа варіантів проекту, і необхідної деталізації на наступному кроці, тому що $\Pi_k \subseteq M_{U_k}, 0 \leq k \leq s-1$. В тому випадку, коли на кожному рівні синтезу параметрів визначений один глобальний критерій ефективності $F_k, 0 \leq k \leq s$, приймає вигляд:

$$\begin{aligned} \Pi_k &= \text{Arg} \max_{m_U(k) \in f_{k+1}^{-1}(\Pi_{k+1})} F_k(m_U(k)) \quad (k = s-1, s-2, \dots, 1, 0), \\ \Pi_s &= \text{Arg} \max_{m_U(s) \in M_{U_s}} F_k(m_U(s)) \end{aligned} \quad (2.6)$$

Рішення (2.6) вимагає застосування методів глобальної оптимізації. Процес рішення (2.5) та (2.6) можна інтерпретувати як синтез СТО, котрий відповідає руху проекту знизу догори, тобто $m_U(s)$ до $m_U(0)$. Інакше, $m_U(0) \rightarrow M_{U(s)}$ описує процес декомпозиції.

Оскільки описана декомпозиція неоднозначна (синтез може бути проведено неефективно), наприклад, невдало обрані f_k і E_k , та може здатися, що множина Π_0 , отримана в результаті рішення рівняння процесу «внутрішньої» розробки, не буде містити жодного проекту з множини $\Pi(E_0, M_{U_0})$, що небажано. Тому, складові f_k та критерії E_k на кожному кроці повинні бути визначеним чином узгоджені.

Для цього використовується поняття узгодженості (якщо для $\forall \bar{m}, \bar{l} \in M_{U_{k+1}}, m_U \in M_{U_k}$ таких, що $\bar{l} B_{k+1} \bar{m}_U, m_U \in f_{k+1}^{-1}(\bar{m}_U)$, знайдеться точка $l \in f_{k+1}^{-1}(\bar{l})$, домінуюча m_U , тобто $l B_k \bar{m}_U$, де $E_k(m_U(k)) = (E_k^1(m_U(k)), \dots, E_k^{n_k}(m_U(k)))$ векторний критерій, визначений на множині M_{U_k} , при фіксованому $k, 0 \leq k \leq n$, B_k – бінарне відношення таке, що для $\forall \bar{m}_U, \bar{l} \in M_{U_k}$ вбачаємо $m_U B_k l$ тоді й тільки

тоді, коли $E_k^i(m_U) > E_k^i(l)$ для усіх i , $1 \leq i \leq n_k$ та знайдеться i_0 , для котрого $E_k^{i_0}(m_U) > W_k^{i_0}(l)$ та цілком узгоджена (якщо вони узгоджені та для k , $0 \leq k \leq n-1$ відображення $f_{k+1}(\bullet)$ ізотонне: для $\forall m_U, l \in M_{U_k}$ з $l B_k m_U \Rightarrow f_{k+1}(l) B_{k+1} f_{k+1}(m_U)$ відношення $B(B_0, B_1, \dots, B_m)$ та синтезу f_k .

Ці визначення встановлюють узгодженість відношень переваг і синтезу, та означають, що розробники k -го та $k+1$ -го рівнів мають однакові уявлення про те, що таке успішний процес створення узагальненої моделі СТО. Точніше, розробник k -го рівня, деталізуючи проект, отриманий від розробника $(k+1)$ -го рівня, може зберегти результат порівняння двох проектів на більш високих рівнях «внутрішньої» розробки.

Повне узгодження означає, що будь-які дві узагальнені моделі СТО, одна з котрих «ефективніше» на k -му рівні, залишаться такими ж і для розробника $(k+1)$ -го рівня. Іншими словами, при переході від одного рівня синтезу до іншого (вищого) не з'являються додаткові критерії оцінки узагальненої моделі СТО, що змінюють результат порівняння двох проектів в ієрархічних системах створення СТО.

Припущення про повну узгодженість відносин переваг розробників і синтезу є природним, оскільки результат порівняння двох ІТ при докладному і грубому (синтезованому) описі повинен залишатися незмінним. Очевидно, що при створенні ІТ слід прагнути до повної узгодженості, тому що вибір E_k , f_k не обмежений традиціями створення СТО і звичками розробників (створюються вперше). У зв'язку з цим важливим є завдання отримання достатніх умов і ефективних алгоритмів перевірки узгодженості відносин і синтезу при заданих E_k , f_k .

На основі теореми [102]: якщо відношення B та синтез f узгоджені, тоді рішення Π_0 рівняння (2.5) містить рішення задачі створення узагальненої моделі СТО (2.4), тобто:

$$(E_0, M_0) \subseteq \Pi_0 \quad (2.7)$$

Запропонований вище підхід дозволяє визначити шляхи формалізації двох основних ситуацій, які мають місце при побудові моделей СТО: послідовного ускладнення і уточнення (розробка "зверху донизу"); послідовного спрощення (розробка "знизу догори").

Неформальний аналіз першої ситуації, в інтерпретації побудови моделей СТО в термінах блок-схем або схем-моделей, показує, що ці поняття лише розкривають його структуру, але не вирішують питання про зв'язки між субоб'єктами, тобто не дозволяють розкрити структуру всієї множини субоб'єктів моделей СТО.

Зв'язки субоб'єктів моделей СТО між собою є дуже важливою обставиною, оскільки зв'язки СТО з субоб'єктами (з яких будується СТО) по суті лежать в основі методу розробки.

Більш точно це можна висловити так: процес розробки СТО по суті є побудовою його базових субоб'єктів за допомогою композицій. Тобто, проблема уточнення структури СТО по суті зводиться до проблеми знаходження базису і композицій.

Очевидно, що основна проблема тут: знаходження композицій (розробка СТО – це створення з заданих базових елементів більш складних елементів) [103, 104]. Очевидно, що якщо строго задані композиції, то, отже, строго задані і декомпозиції, тобто кошти, які дозволяють розкладати більш складні системи на прості.

Технологія паралельного проектування PLM [17, 29] забезпечує одночасний доступ усіх учасників до проекту, реалізованого в ІС PLM. Наявна схема дерева проекту СТО, доповнена в межах розробленого методу введенням складової "Керування параметрами" та внесення іншого наповнення Технологічної моделі, яка пов'язана з іншими моделями СТО, представлена на рис. 2.6.

На рисунку позначено: КС – кінематична схема, СК – схема конструкції, ССО – схема систем і обладнання, СУ – силова установка.



Рисунок 2.6 – Схема дерева проекту складного технічного об'єкта в IC PLM

Ймовірність того, що даний компонент структури знаходиться на i -му рівні, дорівнює P_i .

Позначимо t_i – загальний час доступу до даних, розташованих на i -му рівні. Тоді середній час доступу t_{cep} до компонентів незбалансованої ієрархічної структури проекту СТО, яка складається з N рівнів, дорівнює:

$$t_{cep} = \sum_{i=1}^N P_i t_i$$

Математична модель для визначення загального часу ведення проекту створення узагальненої моделі СТО в IC PLM:

$$t_{cto} = \sum_{j=1}^k \left(\sum_{i=1}^n t_{ji} \right),$$

де: t_{ij} – час створення узагальненої моделі СТО на i -му етапі;

n – кількість етапів створення узагальненої моделі СТО;

k – кількість циклів створення узагальненої моделі СТО.

На рис. 2.7 представлено схему узагальненої моделі СТО.

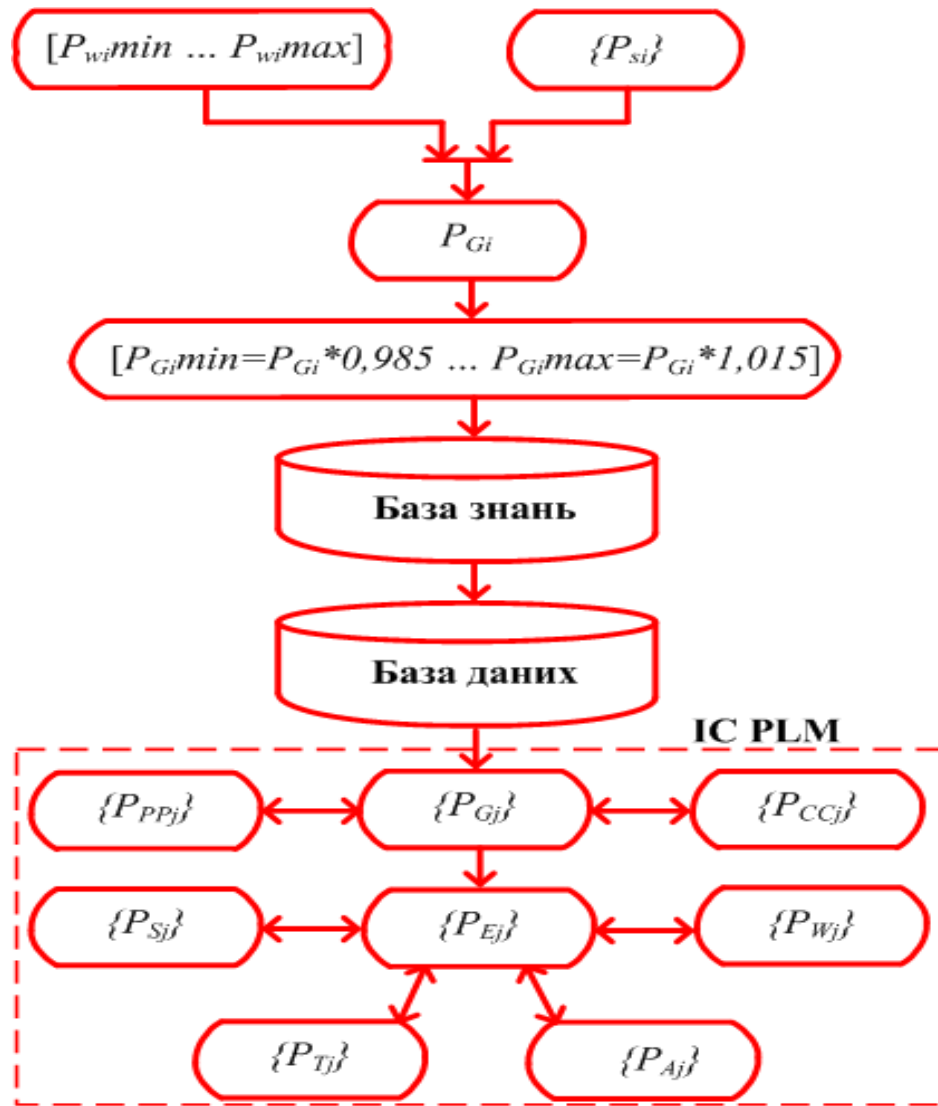


Рисунок 2.7 – Схема узагальненої моделі складних технічних об'єктів

Розглянуто задачу побудови залежності параметрів:

геометричної моделі: $m_{Gi} - p_{Gi}$;

моделі міцності: $m_{Si} - p_{Si}$;

параметрів вагової моделі: $m_{wi} - p_{wi}$ компонента СТО.

На етапі створення БКС i -го компонента СТО в умовах невизначеності важко оцінити міжмодельний вплив.

У зв'язку з чим доцільним є використання для оцінки моделей СТО лінгвістичних змінних (зв'язки $\sigma_{M_{U1}}$).

Терм-множину змінної

p_{wi} позначимо: $P_{wi} = \{P_{wi}min, P_{wi}av, P_{wi}max\}$;

p_{Si} позначимо: $P_{Si} = \{P_{Si}min, P_{Si}av, P_{Si}max\}$;

p_{Gi} позначимо: $P_{Gi} = \{P_{Gi}min, P_{Gi}av, P_{Gi}max\}$,

де представлено множину значень змінної:

$P_{i}min$ – «умовно припустимі до мінімального значення»;

$P_{i}av$ – «припустимі»;

$P_{i}max$ – «умовно припустимі до максимального значення».

Цільова функція визначення УМ має вигляд:

$$\min F(T_{i=1}^m) = \begin{cases} M_G \left[\sum_{i=1}^n \min \left\{ \sum_{j=1}^n \cup_{ij}^{M_G} P_{ij}^{M_G} \right\} \right] \\ M_W \left[\sum_{i=1}^n \min \left\{ \sum_{j=1}^n \cup_{ij}^{M_W} P_{ij}^{M_W} \right\} \right] \\ M_S \left[\sum_{i=1}^n \min \left\{ \sum_{j=1}^n \cup_{ij}^{M_S} P_{ij}^{M_S} \right\} \right] \end{cases}$$

В роботі розглянуто модель Мамдані, що найчастіше використовується, та містить у консеквентах правил логічного висліду нечіткі значення.

Функції приналежності термів обрано відповідно до економічно обґрунтованих значень обмежень вагової моделі та значень моделі міцності.

Створена база правил при побудові нечіткої моделі та в результаті отримуються розраховані обмеження ГМ:

1). **IF** p_{wi} **IS** $P_{wi}min$ **AND** p_{si} **IS** $P_{si}min$ **THEN** p_{Gi} **IS** $P_{Gi}min$

...

9). **IF** p_{wi} **IS** $P_{wi}max$ **AND** p_{si} **IS** $P_{si}max$ **THEN** p_{Gi} **IS** $P_{Gi}max$

Узагальнена модель СТО реалізується за допомогою розробки методу знання-орієнтованого створення СТО, що є базою для розробки ІТ та доповнює і розширює наявний метод інтегрованого моделювання і комп'ютерного проектування введенням рівня керування параметрами моделей СТО з бази даних та знань для зв'язку моделей СТО та проектування і виробництва СТО.

2.4 Розробка методу знання-орієнтованого створення складних технічних об'єктів

На основі проведеного SWOT-аналізу та проведеного аналізу методів та інструментальних засобів розробки та супроводу програмного забезпечення (ПЗ) та складових IT CALS та PLM-рішень, наведено діаграму діяльності методу знання-орієнтованого створення СТО, який задовольняє експертів та усуває недоліки наявного методу та є базою для розробки ІТ.

Аналіз предметної області представляє собою особливий вид наукової діяльності, в результаті якої будується інтерпретаційна модель предметних знань.

Ефективність створення СТО значною мірою обумовлена використанням системного підходу у вигляді таких його складових, як системотехніка і системний аналіз.

Системотехніка – наука, що розробляє методологічні основи розробки складних технічних систем. [106]

Системний аналіз (СА) – науково-практична дисципліна, яка спрямована на розробку методів дослідження або розробки складних систем, а також шляхів та засобів вирішення складних проблем прикладного характеру, які базуються на методологічних принципах системного підходу. [107, 59]

При СА реальний або уявний об'єкт поділяється на складові факти і потім досліджуються ці елементи та зв'язки між ними. [108]

Методи структурного системного аналізу (ССА) успішно застосовуються для вирішення різноманітних задач, наприклад, при управлінні фінансами, плануванні виробництва, організації матеріально-технічного забезпечення, побудови систем діагностики, організації системи освіти та ін., а також при автоматизації проектування та виробництва.

Методи СА описані серією стандартів IDEF (integrated computer-aided manufacturing DEFinition) – методологія функціонального моделювання [109], що використовуються для вирішення завдань моделювання складних систем та

дозволяють відображати і аналізувати моделі діяльності широкого спектру складних систем в різних розрізах.

Технологія структурного аналізу та проектування стандартизована в США як стандарт IDEF0. Методи ССА є основою технології CASE (Computer-Aided Software Engineering, автоматизація процесів проектування та розробки програмного забезпечення).

За допомогою методології IDEF5 онтологія предметної області може бути описана за допомогою певного словника термінів і правил, на підставі яких можуть бути сформовані достовірні твердження про стан даної системи в деякий момент часу. [110]

Наявний метод інтегрованого проектування та параметричного моделювання агрегатів СТО, який використовується наразі при створенні СТО та підсумовує напрацювання провідних вітчизняних та зарубіжних вчених і компаній представлено структурно-функціональною схемою на рис. 2.8 [31].

На рис. 2.9 представлено метод знання-орієнтованого створення СТО, який доповнює та розширює існуючий метод інтегрованого моделювання і комп'ютерного проектування введенням рівня керування параметрами моделей СТО з бази даних та знань для зв'язку моделей СТО, проектування і виробництва СТО, у вигляді діаграми діяльності.

На думку спеціалістів в області СА [59; 111 – 114] для вирішення задач аналізу та проектування деякого об'єкта моделюються:

- функції цього об'єкта, наприклад, за допомогою діаграм потоків даних – DFD (Data Flow Diagrams);
- відношення між даними, які в ньому використовуються, наприклад, за допомогою діаграм "сутність-зв'язок" – ERD (Entity-Relations Diagrams);
- поведінку об'єкта (події), наприклад, за допомогою діаграм діяльності (Activity diagrams).

За допомогою DFD-діаграм зображуються функції (процеси), сховища даних та потоки, що їх пов'язують. [111, 115]

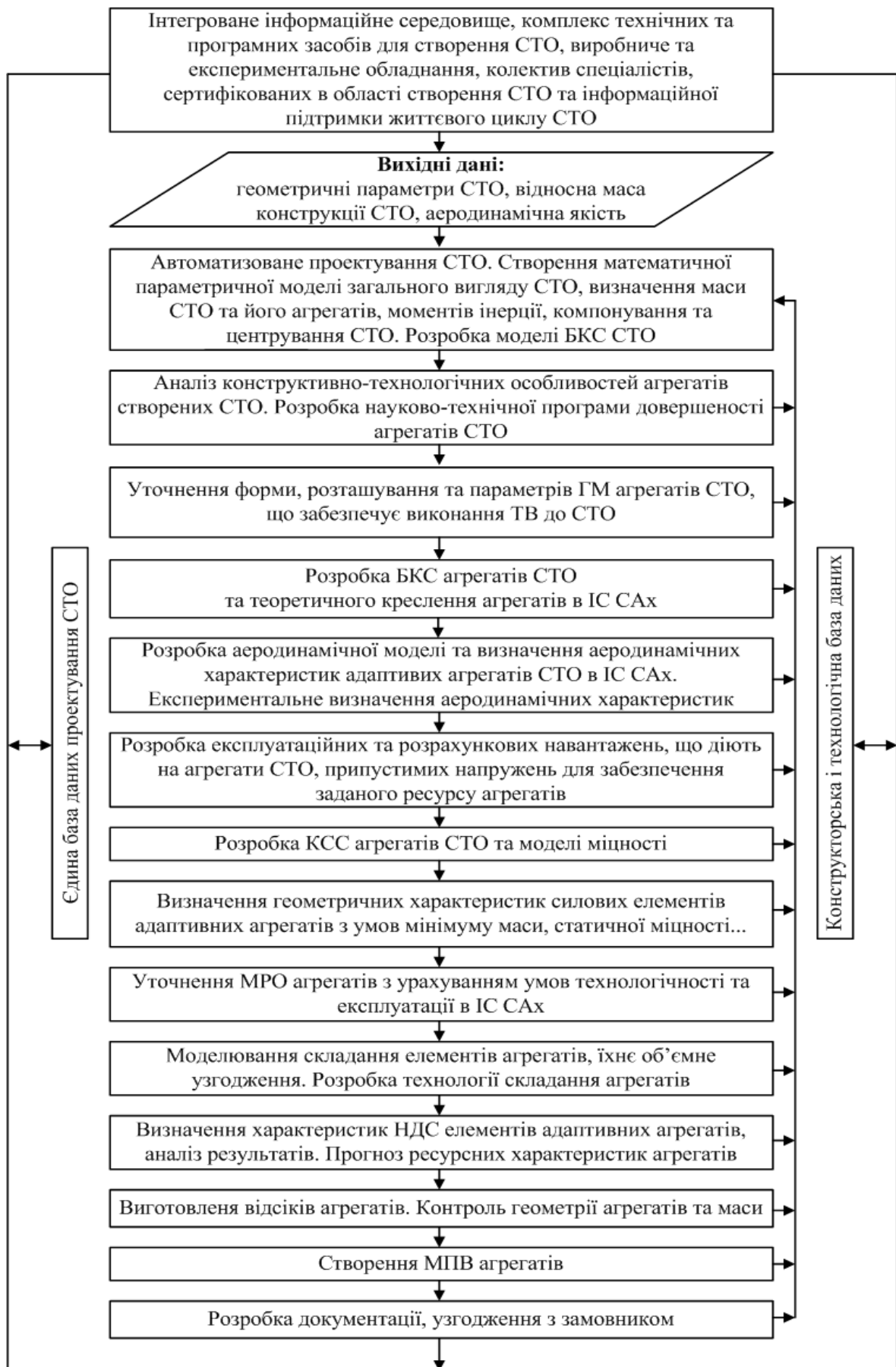


Рисунок 2.8 – Структурно-функціональна схема методу інтегрованого проектування та параметричного моделювання агрегатів СТО

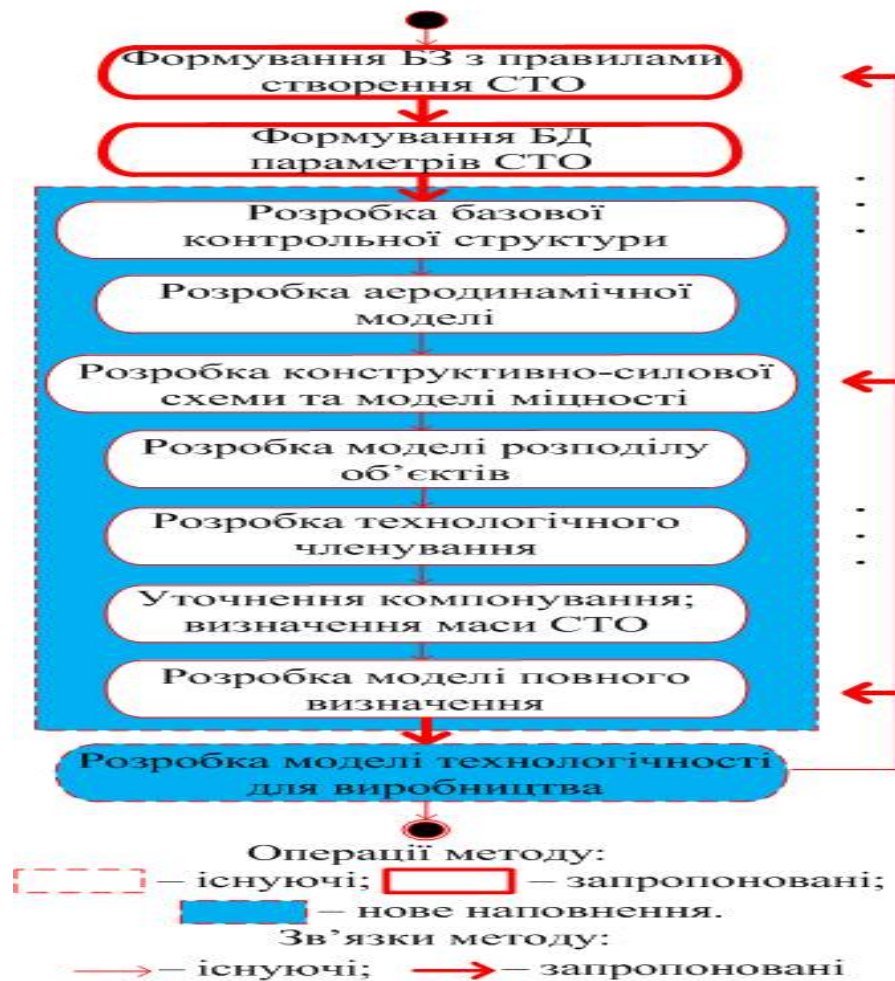


Рисунок 2.9 – Метод знання-орієнтованого створення складних технічних об'єктів

Представлення функцій об'єкта, як правило, відбувається на декількох рівнях деталізації, при цьому на кожному наступному рівні розкривається зміст процесів, показаних на попередньому. На рис. 2.10 представлена контекстна DFD-діаграма методу знання-орієнтованого створення СТО. Представлено DFD-діаграми методу знання-орієнтованого створення СТО: на рис. 2.11 – першого рівня; на рисунках 2.12 – 2.14 – другого рівня.

Побудовані DFD-діаграми дозволяють визначити основні етапи роботи з впровадження ЗОТ в процес створення моделей СТО, а також основні види наборів даних, з якими буде відбуватися робота в процесі функціонування системи: параметри; вихідні дані та результати створення моделей; терміни класифікацій, співвідношення, правила інтерпретації, нормативи, вимоги стандартів.

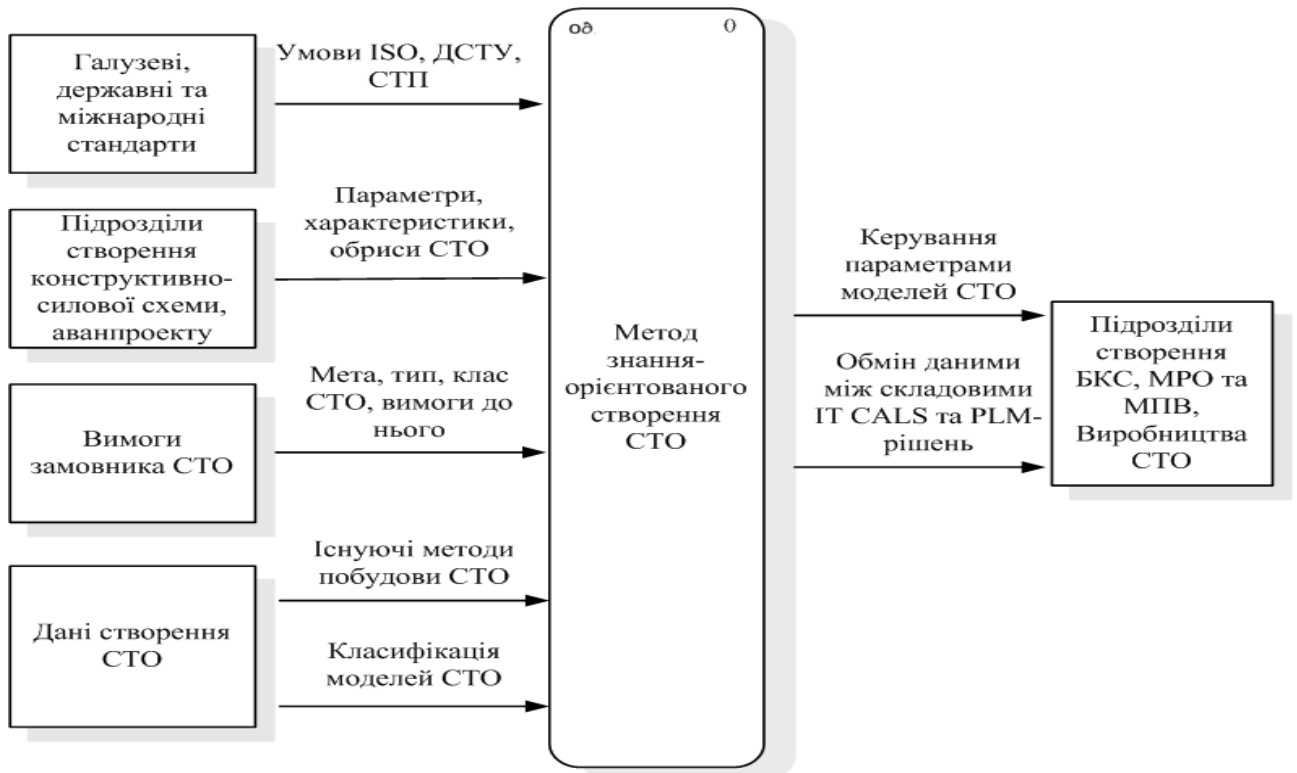


Рисунок 2.10 – Контекстна DFD-діаграма методу знання-орієнтованого створення складних технічних об'єктів

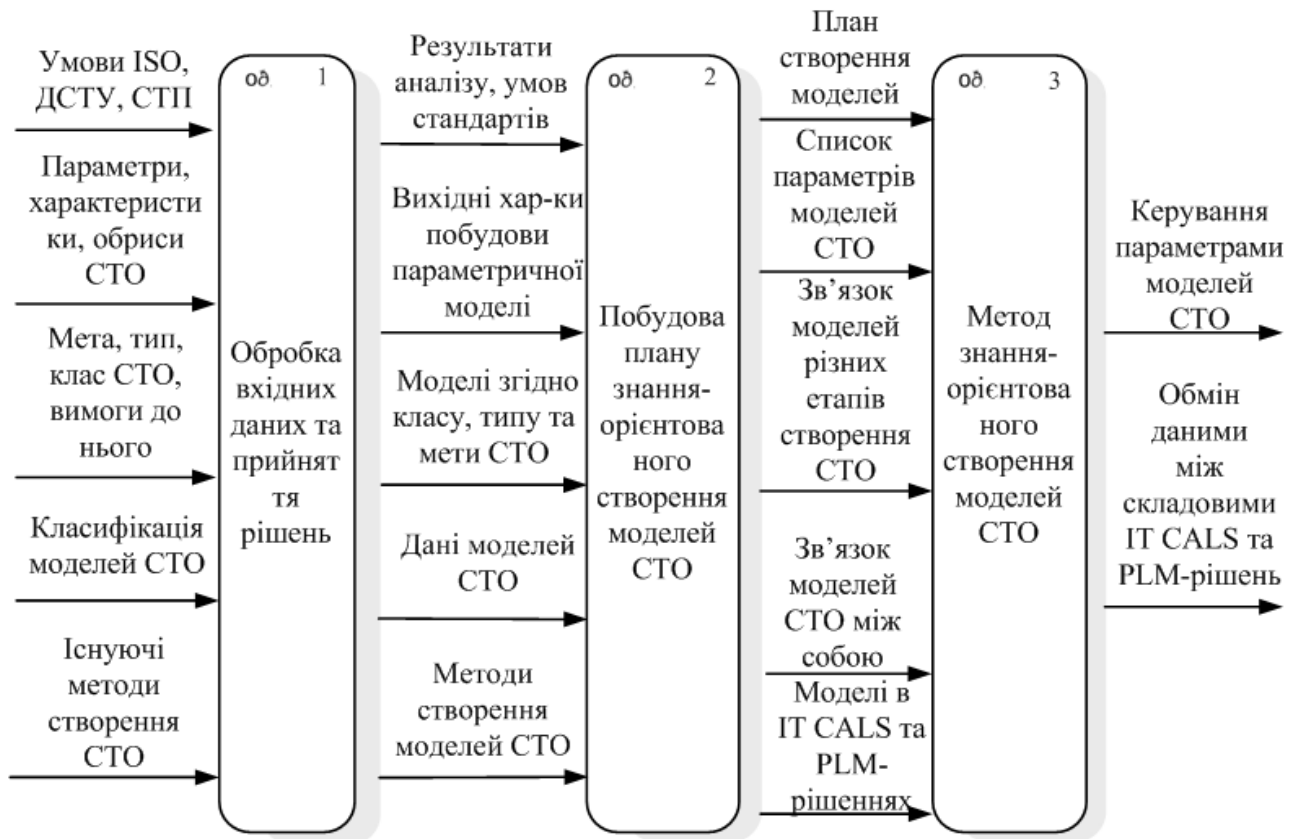


Рисунок 2.11 – DFD-діаграма методу складних технічних об'єктів першого рівня

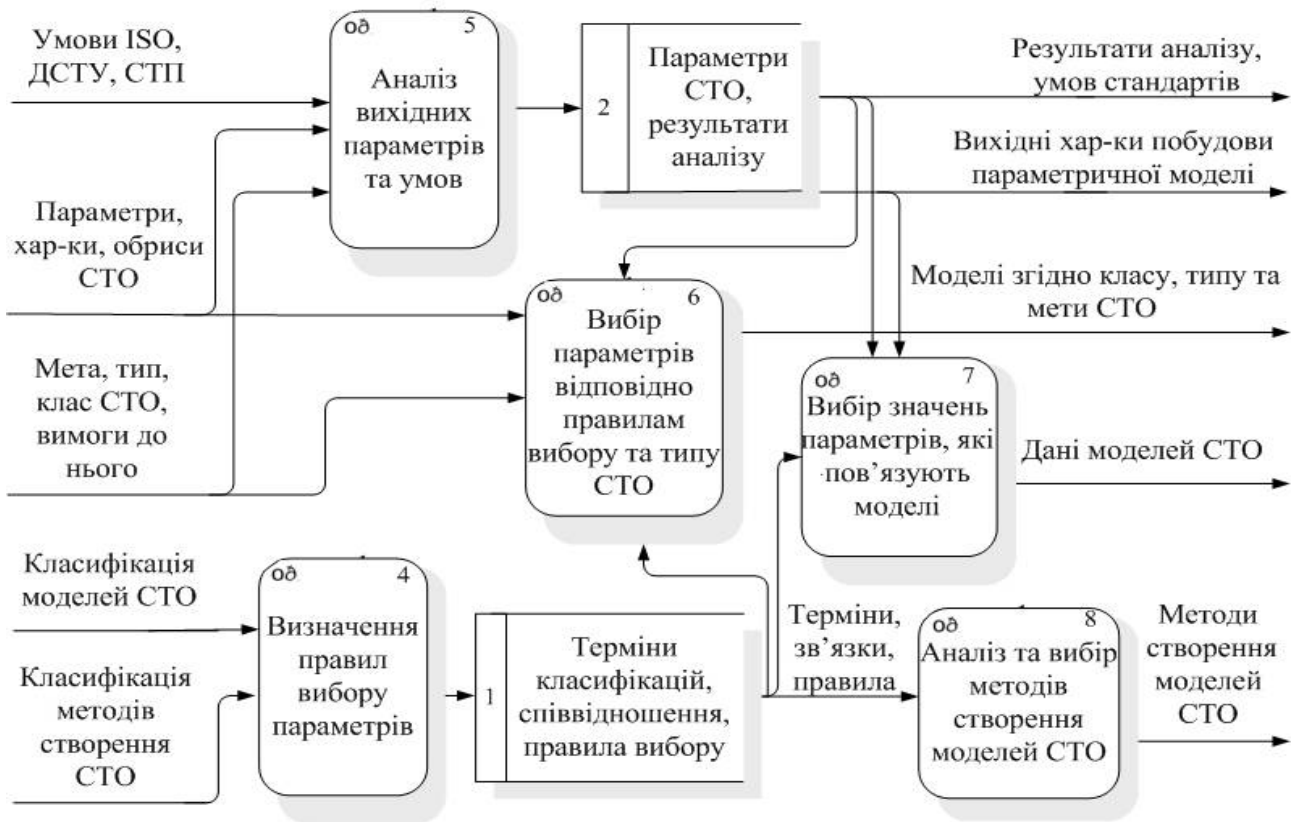


Рисунок 2.12 – DFD-діаграма методу другого рівня

(Обробка вхідних даних та прийняття рішень)

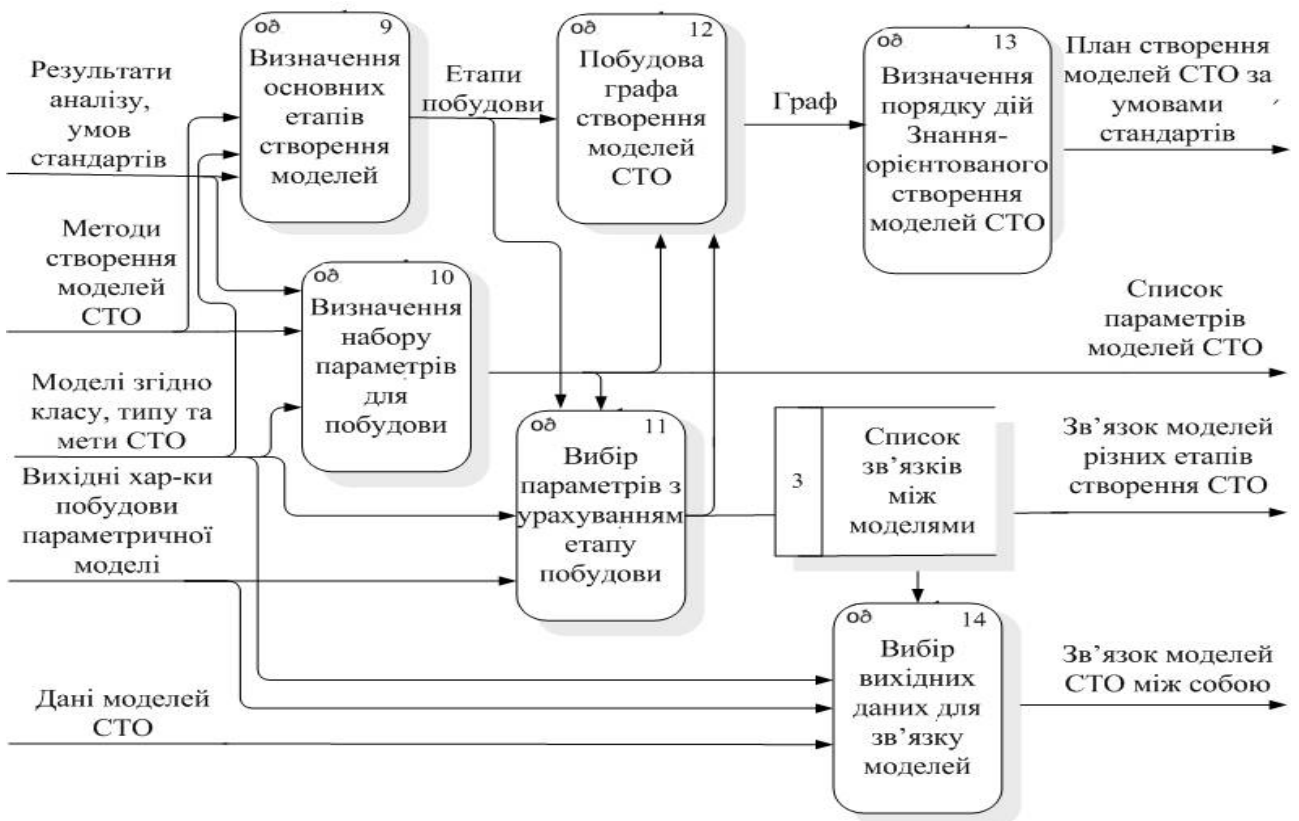


Рисунок 2.13 – DFD-діаграма методу другого рівня

(Побудова плану знання-орієнтованого створення моделей)

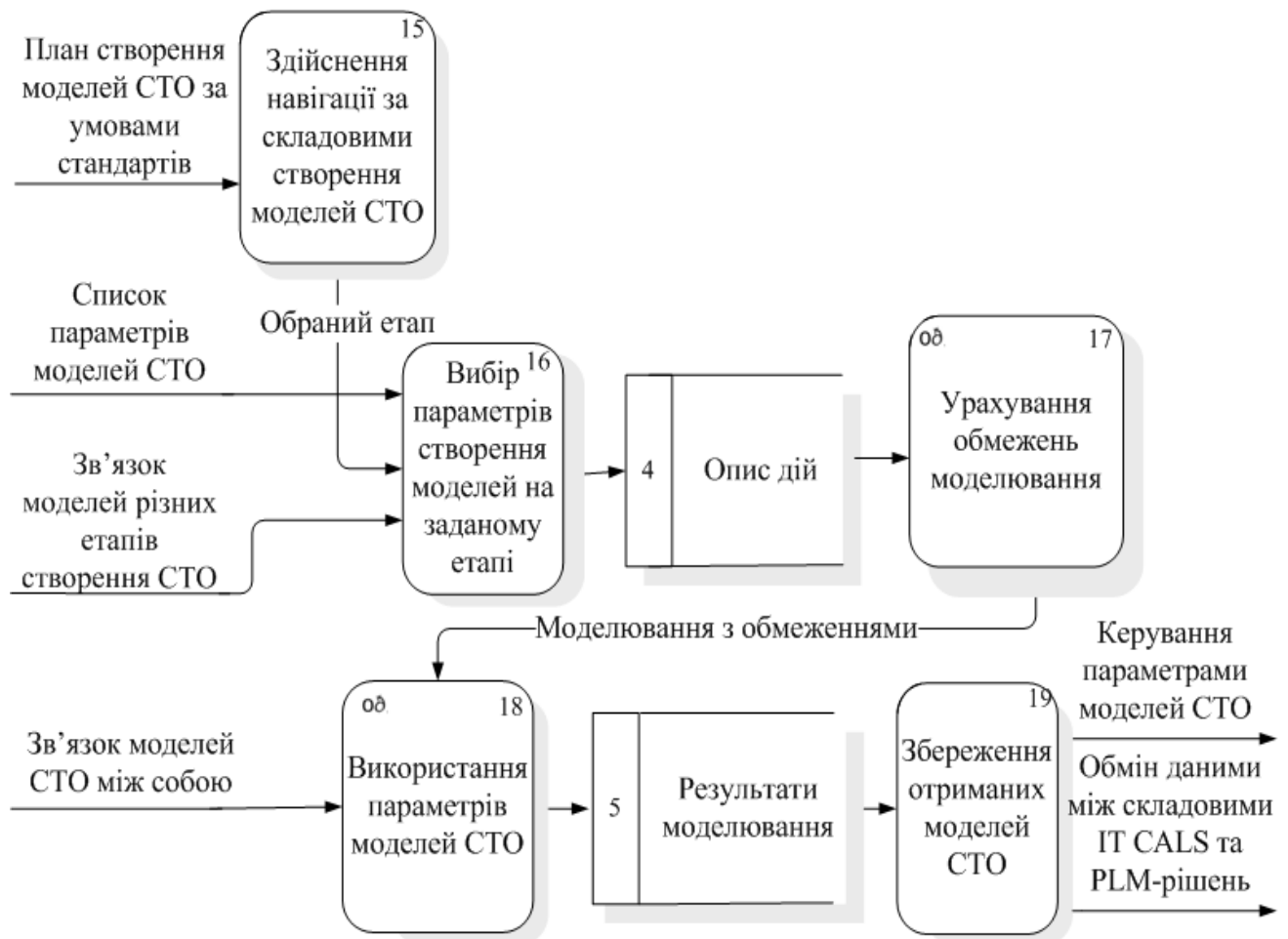


Рисунок 2.14 – DFD-діаграма методу другого рівня

(Знання-орієнтоване створення моделей складного технічного об'єкту)

Для роботи з наведеними вище наборами даних необхідно організувати сховища даних. Діаграми "сутність-зв'язок" (ERD) призначені для розробки моделей даних і забезпечують стандартний спосіб визначення відносин між ними. [111, 116] Дані знання-орієнтованого створення СТО організуються в три структури:

- базу знань (БЗ), що містить терміни з області побудови моделей СТО, зв'язки між ними, правила, що встановлюють порядок інтерпретації термінів, застосування знань, умови проведення розрахунків, узгодження з вимогами стандартів;

- базу даних підприємства, що містить моделі СТО, результати розрахунків, різну службову інформацію та відповідає веденню на підприємстві даних в ІС САх/PLM;

- файлової БД, що містить параметри для визначення моделей СТО.

На рис. 2.15 представлена ERD-діаграма основних наборів даних методу.

Детальніше структура основних сховищ даних знання-орієнтованого створення СТО проаналізована в розділах 3 та 4.

Діаграма діяльності дозволяє доповнити аналіз особливостей знання-орієнтованої побудови в процес створення моделей СТО, показуючи тимчасову послідовність подій у процесі функціонування системи.

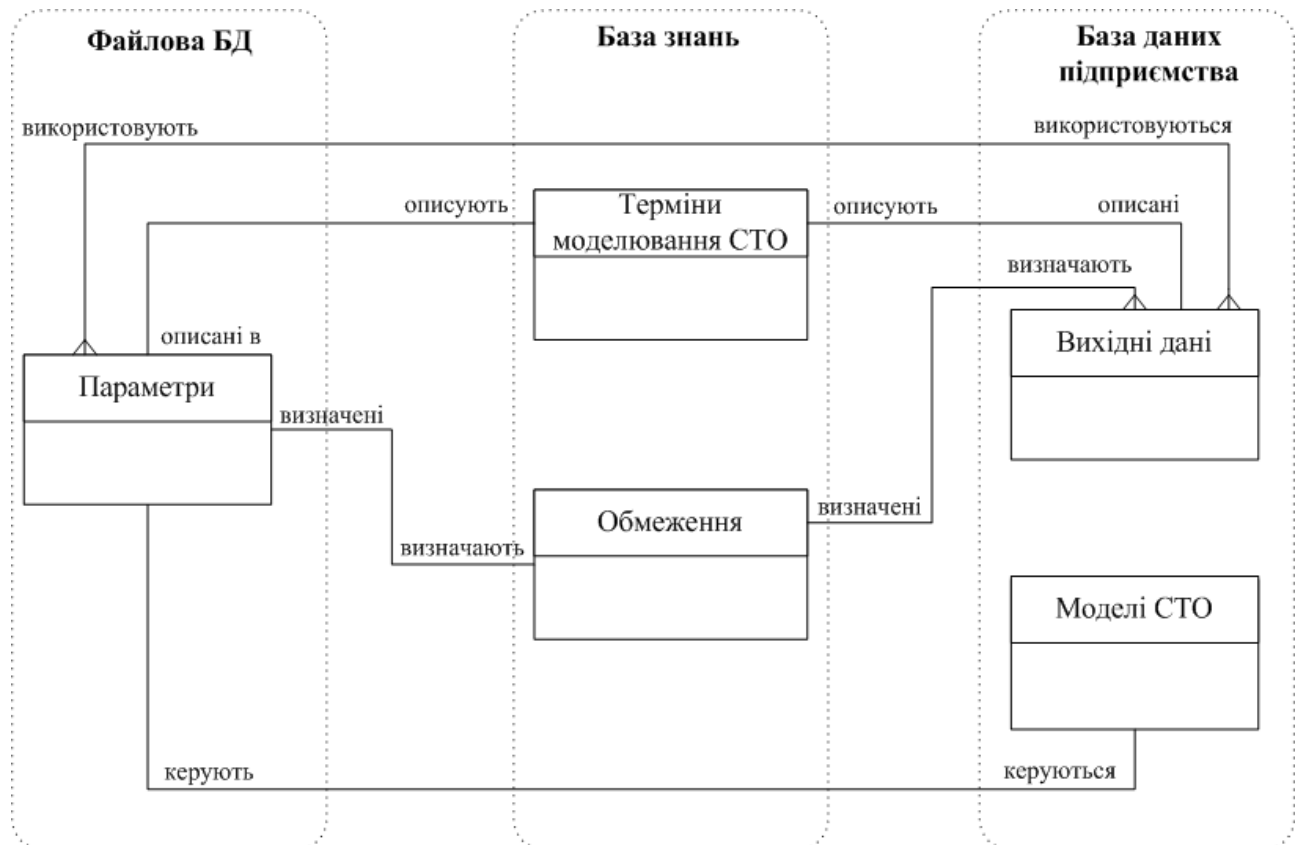


Рисунок 2.15 – ERD-діаграма наборів даних методу знання-орієнтованого створення складних технічних об'єктів

Метод розкриває процес використання параметрів з БД на етапах розробки базової контрольної структури, моделі розподілу об'єктів та моделі повного визначення СТО.

Проаналізувавши основні завдання, вимоги, особливості функціонування та порядок виконання операцій створення моделей СТО, можна сформулювати загальні особливості знання-орієнтованого створення моделей СТО:

- робота з великою кількістю даних;

- ітераційний характер створення моделей на різних етапах створення СТО;
- проведення моделювання СТО за різними методами, вибір яких залежить від типу, конфігурації та параметрів СТО;
- необхідність підбору та використання частинних параметрів та характеристик створення моделей СТО, які будуть використовуватися, починаючи від попередніх етапів створення СТО, і закінчуючи виробництвом СТО;
- необхідність забезпечення зв'язку між різноманітними моделями СТО.

Метод знання-орієнтованого створення СТО включає розв'язок наступних задач:

- 1). вирішення проблеми відстеження зв'язків даних на різних етапах побудови моделей СТО;
- 2). поєднання складових IT CALS та PLM-рішень.

Для розв'язання першої задачі розробляється алгоритм для керування параметричною інформацією.

Для розв'язання другої задачі розробляється алгоритм для обміну даними між різними складовими IT CALS та PLM-рішень.

Вирішення завдання повторного використання знань в загальному випадку складається з двох завдань: надання простого і зрозумілого способу фіксації (стандартизації) тих чи інших типових рішень, та забезпечення доступу до цих типових рішень усім учасникам розробки СТО.

Накопичення та повторне використання знань є одним з тих напрямків роботи зі складовими IT CALS та PLM-рішень, яке робить істотний вплив на скорочення термінів проектування виробу і на підвищення його якості. Ефективне поширення типових конструкторських рішень і використання старих напрацювань допомагає знизити трудовитрати на рутинні роботи і підвищити якість виготовлення виробів.

[117]

2.5 Висновки

1. Представлено концепцію інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів із впровадженням знання-орієнтованих технологій при побудові моделей складних технічних об'єктів, що дозволяє описати побудову складних технічних об'єктів.

2. Досліджено та розроблено узагальнену модель складних технічних об'єктів, що складається з набору основних моделей, отриманих з різних інформаційних систем на різних етапах життєвого циклу складних технічних об'єктів.

3. Представлено проблеми оптимізації процесу моделювання складних технічних об'єктів, що представляє математичне забезпечення процесу створення узагальненої моделі складних технічних об'єктів.

4. Розроблено метод знання-орієнтованого створення складних технічних об'єктів після проведеного аналізу основних особливостей предметної області створення складних технічних об'єктів.

3 РОЗРОБКА ЗАСОБІВ ЗАБЕЗПЕЧЕННЯ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ТА СУПРОВОДЖЕННЯ УЗАГАЛЬНЕНОЇ МОДЕЛІ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ

Проблемна орієнтація та спеціалізація ІТ визначається наявністю уніфікованого методичного забезпечення, орієнтованого на конкретну предметну область, і реалізацією в ньому схеми інформаційної підтримки кожного етапу методу знання-орієнтованого створення СТО.

В розділі розглянуті питання формування уніфікованого методичного забезпечення, яке розширює функціональні можливості процесу розробки ІТ створення моделей СТО відповідно до виділених завдань проблемної орієнтації. В розділі представлено наступні модифікації методу знання-орієнтованого створення СТО: для керування параметричною інформацією при побудові моделей СТО; обміну даними між ІС та засоби реалізації методу.

3.1 Модифікації методу знання-орієнтованого створення складних технічних об'єктів

Процес побудови моделей на різних етапах створення СТО описано як ітераційний процес, на кожному кроці котрого відбувається уточнення складових моделей, що утруднює простеження тотожності вихідних даних.

Для вирішення названої вище проблемної задачі вводиться керування параметрами в процес проектування СТО, що передбачає взаємозв'язок етапів проектування. Розглянемо, що в загальному являє собою параметрична модель. Параметрична модель СТО – це модель, представлена сукупністю параметрів, які встановлюють співвідношення між геометричними та розмірними характеристиками об'єкту, що моделюється.

Існують наступні типи параметризації: ієрархічна, варіаційна, геометрична (розмірна), таблична.

Сучасні методи використання параметрів моделей описують лише доволі вузьку сферу застосування параметричної моделі [118] або в загальному описують побудову зовнішньої форми СТО у вигляді поверхневої моделі [119].

В роботі [120] розкритий метод параметричного проектування з використанням часткових диференціальних рівнянь, який в загальному описує формування обліку СТО у вигляді контурів та ободів на основі диференціальних рівнянь. Проте в розглянутих роботах не розкривається концепція параметричного керування в цілому, не описано інформаційні потоки процесу керування складовими моделями СТО та зв'язок з етапами створення СТО.

Процес керування параметрами моделей побудований за правилами, які визначають розроблювальний проект, і є основою побудови моделей. (рис. 3.1)

Будь-яку модель компонента конструкції, систем та устаткування СТО можна представити у вигляді наступної функціональної залежності [121, 122]:

$$m_i = f(P_i), \quad (3.1)$$

де P_i – параметри відповідного компонента конструкції, систем та устаткування СТО, яку можна представити наступним чином:

$$P_i = \bigcup_{i=1}^n p_i, \quad (3.2)$$

де: p_i – параметр моделей, зокрема: базові точки прив'язки, базові та направляючі лінії, площини, при необхідності – поверхні, n – кількість всіх параметрів, які впливають на моделі.

Загалом, на основі виразу (3.1), модель конструкції, систем та устаткування СТО можна представити наступним чином: $m_i = f^i(p_i, i = 1 \dots n)$.

Керування параметрами моделей призначено для наступних операцій: створення складу та структури виробу; завдання зовнішніх ободів та внутрішніх об'ємів агрегатів, які використовуються в процесі проектування; резервування внутрішніх об'ємів СТО; попередньої компоновки; взаємної ув'язки компонентів конструкції, систем та устаткування; попереднього кінематичного аналізу рухомих

частин агрегатів, вагового аналізу та аналізу на міцність та інших станів об'єкту проектування; побудови на своїй основі різноманітних етапів побудови моделей.

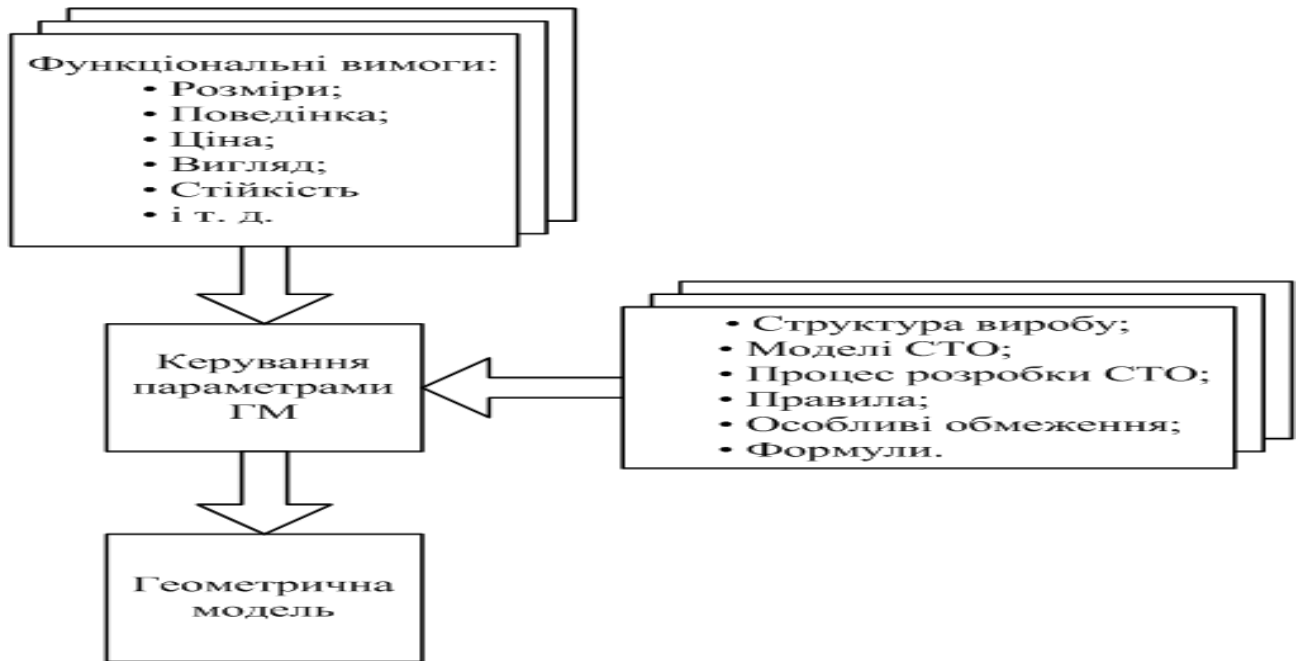


Рисунок 3.1 – Схема керування параметрами моделей

Додатково керування параметрами може включати в себе й інші складові, які необхідні для розробки виробу на відповідному етапі проектування.

Велике значення має процес побудови моделей компонента конструкції, систем та устаткування СТО етапів створення МРО та МПВ, а також БКС, в якій усі параметри повинні бути пов'язані між собою таким чином, щоб при зміні значення головного параметру структура моделі не руйнувалася, а змінювалися лише її габарити. Процес розробки моделей компонента конструкції, систем та устаткування СТО з керуванням параметрами моделей представимо у вигляді орієнтовного графу (рис. 3.2) та діаграми діяльності, використовуючи позначення мови UML [15] (додаток А), де описано:

Блок 1 – Початок розробки.

Блок 2 – Відбувається аналіз технічних рішень по моделі, яка розробляється та визначається які моделі БКС (КС, СК, СРСО) і моделі МРО та МПВ будуть розроблюватися та які параметри з БД будуть використовуватися.

Блок 3 – Якщо модель компонента, який розробляється, має рухомі частини, тоді переходимо на блок 4, в протилежному випадку – переходимо на блок 10.

Блок 4 – Розроблюється модель КС агрегату виробу.

Блок 5 – Розроблюється модель СК агрегату виробу.

Блок 6 – Розроблюється модель СРСО агрегату виробу.

Блок 7 – Якщо зв'язок моделі СК та СРСО відіграється з моделлю КС, тоді переходимо на блок 8, у протилежному випадку – переходимо на блок 4.

Блок 8 – Розробляється модель використовуючи КС, СК та СРСО на етапі МРО.

Блок 9 – Якщо розроблена модель відповідає заданим вимогам, то процес розробки закінчується, у протилежному випадку переходимо на блок 2.

Блок 10 – Розроблюється модель СК агрегату виробу.

Блок 11 – Розроблюється модель СРСО агрегату виробу, перехід на блок 8.

Блок 12 – Закінчення розробки.

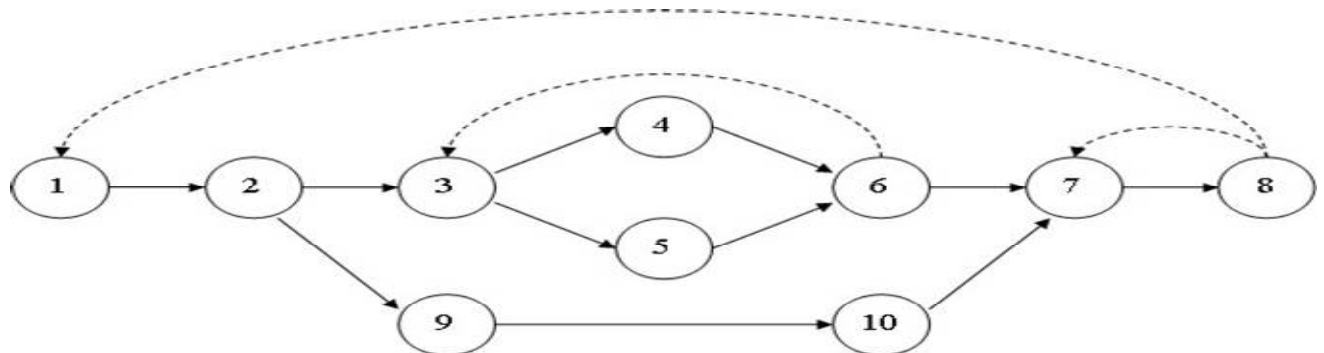


Рисунок 3.2 – Граф керування параметрами моделей

Для вирішення проблеми відстеження тотожності вихідних даних на різних етапах створення моделей СТО розроблена модифікація використання методу знання-орієнтованого створення СТО для керування параметричною інформацією (рис. 3.3), що передбачає створення взаємозв'язку етапів створення моделей СТО.

Переваги знання-орієнтованої побудови моделей СТО для керування параметричною інформацією, основою якої є онтології:

- Онтологія може представляти кілька типів відносин між об'єктами.
- Онтологія може бути інтегрована з ІС PLM та САХ.

- Онтологія може бути керована з IC PLM та САх.

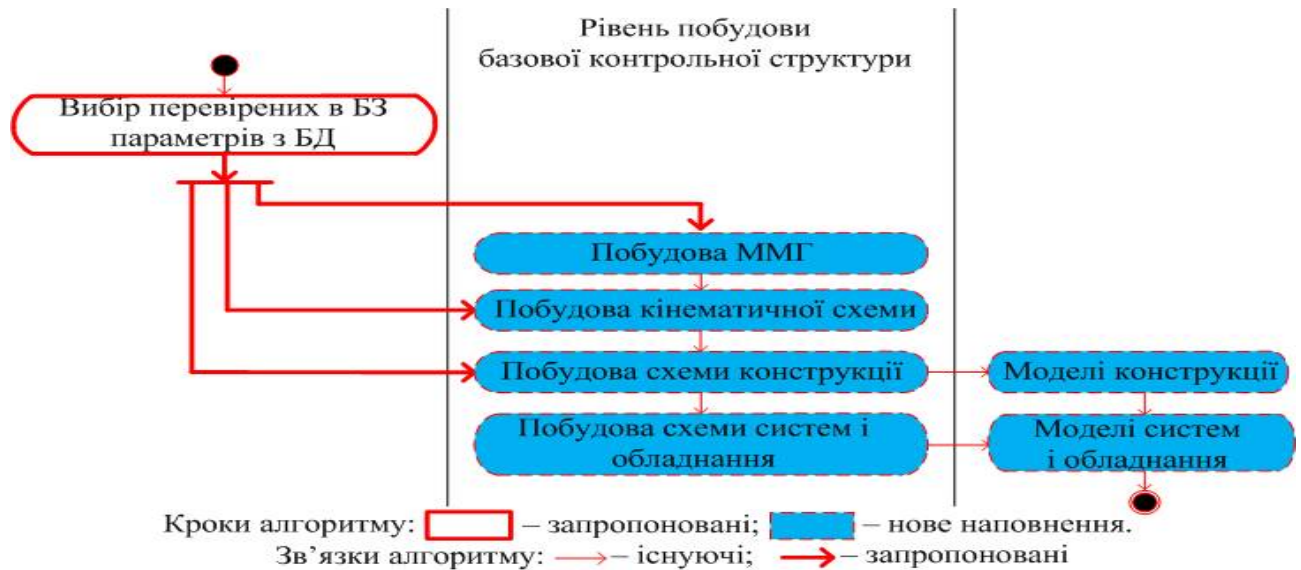


Рисунок 3.3 – Модифікація методу для керування параметричною інформацією

Розглянемо модифікацію методу знання-орієнтованого створення СТО на основі знання-орієнтованих технологій для обміну даними між складовими IT CALS та PLM-рішень. При створенні моделей СТО в різних IC САх виникають недоліки, які пов'язані з обміном даними цих моделей. Надалі пропонується застосування знання-орієнтованих технологій при обміні даними моделей, створених у різних IC PLM та САх.

Моделі СТО створюються за допомогою обраних підприємством або підрозділом підприємства IC САх, але часто виникає необхідність обробки цих моделей в інших IC САх. Втрата даних під час трансляції призводить до того, що модель, передана між різними системами, стає неповною.

Доступні транслятори розроблюються для конкретних IC САх, є неуніверсальними та мають високу вартість: стандарт базової специфікації обміну графікою (IGES) [123]; стандарт для опису даних про виріб (STEP) [124]; формат STL – Stereolithography [125] і т. д.

Прикладний протокол AP203 стандарту (STEP) [126] дозволяє передавати зовнішній контур моделей деталей та складальних одиниць. Стандарт ISO 10303-111 [127] забезпечує представлення операцій для побудови орієнтованих на топологію твердотільних моделей. Основою стандарту STEP є об'єктно-орієнтовна

мова EXPRESS, призначена для опису моделей світу (схем продукції) на концептуальному рівні. [128]

Стандарт WebGL дозволяє створювати інтерактивну графіку, що функціонує в широкому спектрі сумісних з нею веб-браузерів, які підтримують HTML, не вдаючись до посередництва плагінів. [129]

Для розширення можливостей IC САх в динамічному моделюванні використовують пакет Matlab. [130]

Перераховані формати передачі даних між різними IC САх нездатні до обробки та передачі повного набору інформації про моделі, зокрема історії побудови моделей та обмежень всередині неї.

Описано багато методів реалізації обміну даними між складовими IT CALS та PLM-рішень, кожен з яких має переваги та недоліки. Розглянемо найвідоміші з них.

Макро-параметричний метод [131], в якому при передачі параметричної інформації, включаючи історію побудови моделей, ряд стандартних команд визначений і використовується як нейтральний формат, сама ж історія побудови міститься в макро-файлі, який використовується для зразкового обміну. Проте метод не можна застосувати для IC САх, які не роблять макро-файли, він не включає обмін обмеженнями і параметрами.

Метод, заснований на використанні подвійної моделі [132] передбачає розбиття моделей: на модель, яка визначає історію побудови та модель, яка містить представлення моделі у вигляді зовнішнього контуру. Даний метод описує тільки використання обмежень, які визначені в контексті ескізу. Робота також обмежена обміном тільки моделей деталей, а не складальних одиниць.

Метод обміну, що використовує розширювану мову розмітки (XML) [133] для вираження набору стандартних моделюючих команд, отриманих з моделей IC САх передає історію побудови моделей, проте ця історія базована тільки на обмеженому наборі команд моделювання. В методі не розглянуті обмеження.

В роботі пропонується застосування знання-орієнтовних технологій при обміні даними моделей, створених у різних IC САх. Завдяки використанню

онтології створюється можливість автоматичного пошуку залежностей та знаходження конфліктів між складовими побудованої моделі.

Існуючі методи застосування ЗОТ при обміні даними моделей з різних ІС САх мають певні недоліки. Зокрема, робота [134] описує використання дворівневої онтології, в якій онтологія верхнього рівня відіграє подібну роль до стандарту STEP. У роботі [135] визначений набір геометричних обмежень для визначення форми моделей, яка необхідна, щоб дозволити сумісність при обміні даними між різними ІС САх. Автори даних робіт подають тільки інформацію про обмеження, які необхідні при передачі даних, але не розглядають інформацію, як ці обмеження можуть бути передані.

Загальна схема реалізації обміну даними моделями СТО з різних складових ІТ CALS та PLM-рішень, використовуючи ЗОТ, представлена на рис. 3. 4. [136]

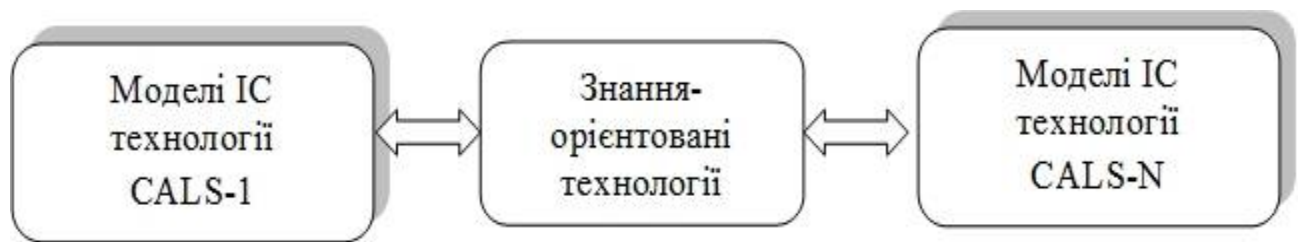


Рисунок 3.4 – Обмін даними моделей складного технічного об'єкта з використанням знання-орієнтованих технологій

Варіант процесу обміну даними моделями з різних складових ІТ CALS та PLM-рішень, використовуючи ЗОТ, представимо у вигляді діаграми діяльності (додаток А).

В роботі розроблено модифікацію використання методу знання-орієнтованого створення СТО для обміну даними між різними складовими ІТ CALS та PLM-рішень (рис. 3.5) – це процес конвертації даних з різних складових ІТ CALS та PLM-рішень.

Наступні розділи розкривають процес обміну даними моделями з різних ІС PLM та САх, використовуючи ЗОТ.

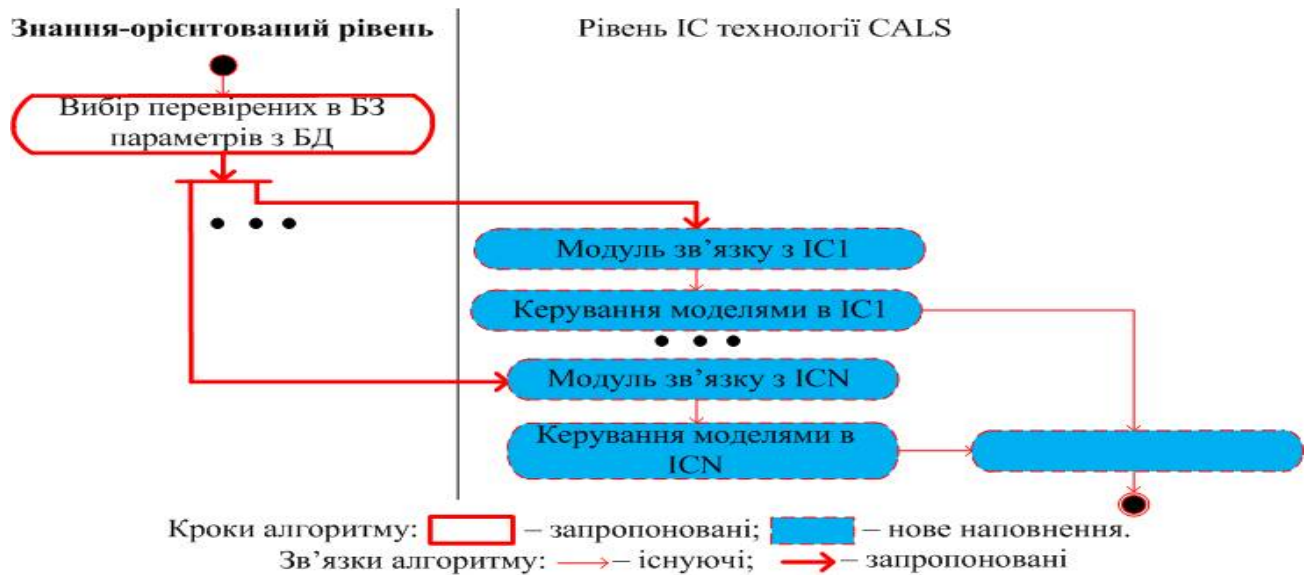


Рисунок 3.5 – Модифікація методу для обміну даними між складовими інформаційних технологій CALS та PLM-рішень

Переваги знання-орієнтованої побудови моделей для обміну даними, основою якої є онтології:

- Можливість перенесення параметрів моделей СТО з ІС PLM та САх системи в онтології.
- Можливість візуального представлення відносин між моделями.
- Можливість візуально представляти причинно-наслідковий ланцюжок змін в моделях.

Звідси витікають можливості розробки ІТ: здатність представляти правила і формули відносин між моделями і параметрами; можливість автоматичного оновлення моделей, де такі правила або формули відображаються.

3.2 Онтологічний опис процесу створення узагальненої моделі складних технічних об'єктів

На основі розробленого методу знання-орієнтованого створення СТО при побудові узагальненої моделі СТО обираються засоби реалізації методу. Узагальнена модель СТО описується причинно-наслідковими зв'язками з БЗ (онтологічна схема, правила створення СТО): σ_{MU2} .

Існують різні моделі БЗ, з різними підходами до представлення інформації предметної області, організації її зв'язності, застосування правил інтерпретації для кожного окремого кванта інформації. Найчастіше застосовуються наступні моделі представлення знань [137, 138]:

- Логічна модель представлення знань: обчислення предикат і обчислення виразів. Використання системи предикат накладає обмеження на інформацію, збережену в БЗ, порядок логічного виводу, систему прийняття рішень. Особливості предметної області побудови моделей СТО вимагають від БЗ більшої гнучкості, тому можна реалізувати використовуючи систему предикат. Реалізація алгоритмів придбання нових знань на основі вирахування предикат вимагає в цьому випадку занадто великі обсяги пам'яті для розв'язку.

- Продукційна модель. Представлення БЗ у вигляді сукупності продукційних правил з обліком усіх можливих випадків і комбінацій параметрів СТО займе багато часу й програмного коду. Тому для представлення БЗ ця модель теж не підходить.

- Фреймова модель. Представити знання про вироби у вигляді фреймів, а знання про складові частини СТО і його параметрів у вигляді слотів з наступним розкриттям цілком можливо. Однак такий варіант моделі БЗ буде мати кілька недоліків: складна загальна структура моделі; невизначена кількість слотів, які відображають складальні одиниці й деталі СТО; складність відображення зв'язків між компонентами СТО, функціями розрахунків, користувачами програми; недостатньо гнучкий механізм прямих і зворотних зв'язків між компонентами.

- Онтологічна модель. Найбільш зручно представити різнорідну інформацію предметної області "Створення узагальненої моделі СТО" у вигляді об'єктів, класів, зв'язків між ними, а також правил інтерпретації. [139]

Застосування онтології допоможе зв'язати між собою такі суттєво різні поняття, як компоненти СТО, маси компонентів, вхідні параметри моделей СТО, а також місце на вінчестері, де зберігається необхідна для роботи системи інформація. Онтологію певної галузі знань разом з відомостями про властивості конкретних об'єктів можна назвати базою знань (БЗ). [140] До сьогоднішнього дня

велику кількість онтологій було побудовано різними групами, для різних цілей, використовуючи різні методи та технології. [141 – 144]

Онтологія за визначенням Грубера – це явна специфікація концептуалізації, абстрактне осмислення та спрощений погляд на поняття і відносини, що воно представляє [140]. Використання онтологій узагальнено в підсумку [145] у такий спосіб:

- для комунікації: між розглянутими обчислювальними системами; між людьми; між людьми й розглянутими обчислювальними системами;
- для обчислювального виводу: для представлення внутрішніх розробок і керування розробками та планування інформації; для аналізу внутрішніх структур, алгоритмів, вхідної й вихідної інформації розглянутих систем теоретично й у концептуальний термін;
- для повторного використання (і організації) знання: для структурування та організації бібліотек як сховища розробок і планів, і області інформації. [89]

Онтологію (O) області створення узагальненої моделі СТО представимо:

$$O = \langle M_U, R, F \rangle,$$

де: M_U – множина концептів (понять, термінів) області створення УМ СТО, R – скінчена множина відношень між концептами (поняттями, термінами) області створення УМ СТО, F – скінчена множина функцій інтерпретації, які задані на концептах і відношеннях онтології. [139, 146]

Розгляд граничних випадків перелічених множин: $R=0$; $R \neq 0$; $F=0$; $F \neq 0$ в усіх чотирьох комбінаціях значень R та F дає різні варіанти онтологічних конструкцій, починаючи від простого словника предметної області та таксономії до високоінтелектуальних знання-орієнтованих систем.

Онтологія предметної області характеризується типами відношень:

$$R = \{R_{P-B}, R_{C-E}, R_{O-A}\},$$

де R_{P-B} – «рід» – «вид»; R_{C-E} – «система» – «елемент» (ціле-частина); R_{O-A} – «об'єкт» – «атрибут». Перший тип визначає різновидність об'єктів (понять), другий – склад (об'єм поняття), третій – сукупність властивостей об'єкта.

Для відношення функцій інтерпретації вірно: $F = F(M_U, R)$, де $\forall_i (m_i \in M_U, m_i = F(M_{U_{i-1}}, R^1))$, $M_{U_i} = M_U \setminus m_i$; $|M_{U_{i-1}}| = |M_U| - 1$; $i = 1 \dots |M_U|$; $|M_U| = \text{Card}\{M_{U_i}\}$; $R^1 \in R$.

Вершини графу побудови онтології $m_i \in M_U, i = \overline{1, \text{Card}M_U}$ розподілені за рівнями, так що: $M_U = \bigcup_i M_{U_i}; i = \overline{0, N}$. На 0-му рівні є лише одна вершина, яка відповідає назві предметної області «Створення УМ СТО», будь-яка вершина наступних рівнів пов'язана з однією вершиною верхнього рівня та визначеною підмножиною вершин нижнього. Множині M_U поставлено у відповідність множину індексів його елементів (вершин графу): $M_U \rightarrow G = \bigcup_i G^i, i = \overline{0, N}$.

Онтологія створення УМ СТО має відповідати наступним вимогам [147]:

- повинна містити концептуальні, а не епізодичні знання і не залежати від природної мови, яка використовується для опису знань;
- повинна бути добре специфікованою і внутрішньо узгодженою із структурою, іменами і вмістом для всіх визначених в ній понять, а також сумісною із різними джерелами (ресурсами) знань;
 - повинна бути структурованою і простою для розуміння і пошуку понять;
 - обмежується конкретно проблемною областю для визначення понять і не повинна містити всю можливу інформацію про наочну область та навколишній світ, а лише ті поняття, які необхідні для даного застосування;
 - повинна запам'ятовувати і надавати інформацію про свої попередні стани по мірі розвитку, а також підтримувати наступні функції: можливість трансляції з однієї мови на іншу, надавати можливість багаторазового використання, бути редагованою, можливість бути доповненою або доповнювати іншу онтологію, забезпечувати можливість сумісного доступу користувачів.

Результатом врахування вищеперерахованих вимог є розроблена онтологія створення узагальненої моделі СТО у вигляді структури, яка відповідає стандарту оформлення онтологій IDEF 5 та представлена на рис. 3.6.

Онтологія створення узагальненої моделі СТО описує основні зв'язки та співвідношення між частинами процесу створення СТО на стадії розробки ЕП, який є основним етапом побудови моделей СТО, сполучною ланкою між ТВ, БКС та МПВ. МРО передбачає компонування конструкції, систем і обладнання відповідно до ММГ. МРО включає стадії ЕП та ТП. [148 – 158]

Процес побудови узагальненої моделі СТО містить чотири основні частини (підкласи проектування): умови, обладнання, геометрична модель і атрибути. І кожен з цих класів має власні підкласи.



Рисунок 3.6 – Онтологія створення узагальненої моделі СТО

- «Умови» – містить основні відомості про використані документи на етапі проектування МРО та зв'язок цього етапу з іншими етапами створення СТО через різні види документів. Зокрема, документація включає такі етапи створення СТО: Стандарти – документи, які використовуються при розробці МРО. Наприклад: ISO, ГОСТ, ОСТ, ДСТУ, довідкова література зі створення СТО; Технічні рішення – містять у собі вимоги до основних технічних даних СТО, як-от: характеристики СТО; очікувані умови експлуатації; основні геометричні характеристики. Включає етапи розробки технічного завдання та технічної пропозиції, що передують розробці

БКС та МРО і пред'являє основні технічні вимоги до подальшого створення СТО; Економічна модель – модель з економічними характеристиками СТО; Аеродинамічна модель – модель, яка містить аеродинамічні характеристики СТО.

- «Обладнання» показує варіанти програмних продуктів та апаратної частини, які використовуються в процесі створення СТО. Клас «Обладнання» має наступні підкласи: «Сервери» та «Інформаційні системи». «Сервери» містять необхідне для забезпечення нормальної мережевої роботи інженерів устаткування, якими займаються системні адміністратори, або відповідальні за дані роботи компетентні конструктори або програмісти; «Інформаційні системи» містять весь необхідний конструкторів-проектувальників апаратний та програмний набір інструментів, який використовується у процесі створення моделей СТО. Зокрема, це розглянуті в розділі 1 сучасні ІС САх: САТІА, NX, ProE під управлінням таких відповідних ІС PLM, як: Enovia, Teamcenter і т. д.

- «Атрибути» моделей описують основні характеристики кожного файлу моделей СТО. Для моделей деталей і складальних одиниць використовується наступний мінімальний перелік атрибутів: N групи; Найменування; Позначення; Маса, кг; X, Y, Z центра мас, мм; Матеріал. Частина цих характеристик задається користувачем, частина, яка стосується вагової моделі та моделі компонування та центрування визначається щодо локальної системи координат побудови моделі деталі за допомогою програмних засобів ІС САх, в якій відбувалося створення моделей.

- «Геометрична модель» містить 3D-моделі відповідно до дерева проекту СТО в ІС PLM та містить повну модель СТО на етапі МРО. Це такі частини: «Керування параметрами», «БКС», «Силова установка» (СУ), «Системи та обладнання» (СО) та «Конструкція». Базова контрольна структура – це математична модель об'єкта проектування з параметрами, при зміні яких відбувається зміна конфігурації моделей; «Конструкція» містить ГМ складальних одиниць СТО і т. д.

Між усіма вищезазначеними класами та підкласами онтології існує зв'язок, який можна побачити з рис. 3.6. Наприклад: клас «Обладнання» «створює» «ГМ».

Тобто, онтологічна схема створення узагальненої моделі СТО містить усі складові компоненти дерева проекту СТО з усіма моделями СТО та їхніми характеристиками.

Онтології реалізуються в таких прикладних програмних пакетах, як: Apollo, Protégé, OntoBuilder, WebOnto, Ontology Editor, OpenCyc, Visual Ontology Modeler, Knowledge Server, Ontolingua, Chimaera та ін. В табл. 3.1 представлені основні програмні продукти, в яких можливе створення онтологій.

Таблиця 3.1 Інструментальні засоби для проектування та підтримки онтологій

Найменування	Функціонал
Створення онтологій	
Ontolingua	Підтримка спільної розробки
WebOnto	Підтримка спільного перегляду
Protégé	Створення, перегляд, підтримка
OntoSaurus	Web-браузер баз знань LOOM
ODE, WebODE	Створення, методологія Methontology
KADS22	Проектування моделей знань по методології CommonKADS
OntoEdit	Розробка та підтримка онтологій
OilEd	Підтримка міркування
I com	Підтримка
Об'єднання та відображення онтологій	
PROMPT	Додаток об'єднання до Protégé
Chimaera	Інструмент об'єднання, на основі редактора онтологій Ontolingua
OntoMerge	Інструмент об'єднання та міркування
OntoMorph	Перетворення символічних знань
OBSERVER	Інформаційна система на основі взаємодії онтологій
FCA-Merge	Метод порівняння онтологій
ONION	Система композиції онтологій
Анотування Web-ресурсів	
SHOE's Knowledge Annotator	Опис змісту Web-сторінок
MnM	Анотація на основі онтологій
Metabrowser	Створення та перегляд метаданих

Інструментальні засоби побудови онтології можна розділити на два типи: розроблені для редагування онтології на якійсь визначеній мові онтології та ті, які інтегруються в інструментальні сайти і більшість з яких не залежить від мови представлення.

Вихідну версію різних онтологій можна, наприклад, імпортувати з бібліотеки Ontolingua [159], у якій використовувалася версія, опублікована у роботі [160] і розповсюджувана із системою подання знань CLASSIC.

Ключовим моментом у проектуванні онтології є вибір відповідної мови специфікації онтології – OWL (Ontology web language) [161] та необхідних для проектування інструментів. Існують традиційні мови специфікації онтології.

Одна з найбільш популярних програм Protege 4.3 описує онтології декларативним образом, визначаючи явно яка класова ієрархія та до яких класів належать індивідні концепти. Protege-4.3 була розроблена групою Марка Мьюсена в Stanford Medical Informatics [162].

3.3 Розробка бази даних та бази знань створення узагальненої моделі складних технічних об'єктів

У реляційній БД дані зберігаються у вигляді прямокутних таблиць (відносини), при цьому всі операції над БД зводяться до маніпуляцій із цими таблицями. [163] Сама БД складається з відносин, зв'язок між якими встановлюється за допомогою співпадаючих полів.

При проектуванні реляційної БД необхідно вирішити питання про найбільш ефективну структуру даних.

Основні цілі проектування БД: [164] забезпечити швидкий доступ до даних у відносинах; виключити непотрібне повторення даних; забезпечити цілісність даних таким чином, щоб при зміні даних відносин автоматично відбувалася відповідна зміна пов'язаних з ними об'єктів.

Дані знання-орієнтованого створення узагальненої моделі СТО організовані в наступні БД (відповідно рис. 2.7):

- базу даних підприємства, що містить моделі СТО, результати розрахунків, різну службову інформацію;

- файловою базу даних, що містить параметри для визначення моделей СТО (БД "Параметри моделей СТО".)

Розглянемо організацію БД підприємства, що містить моделі СТО, результати розрахунків, різну службову інформацію. База даних підприємства знаходиться в ІС PLM [165, 166] містить моделі компонентів деталей, складальних одиниць, агрегатів СТО.

Для цих об'єктів властиві наступні атрибути: шифр СТО, найменування, номер службової записки, ППП розробника, дата завантаження, маса, перелік технічних вимог до побудови компонента.

Характеристики моделей описують основні характеристики (атрибути) кожного файлу моделей. Для моделей деталей і складальних одиниць використовується наступний мінімальний перелік атрибутів: N групи; Найменування; Позначення; Маса, кг; X центра мас, мм; Y центра мас, мм; Z центра мас, мм; Матеріал.

- Довідкові дані. Для одержання довідкової інформації використовуються джерела: нормативи ISO, ДСТУ, ГОСТ, ОСТ, СТП, АТА-100, 57АТ, ММГ, БКС, МРО, дані архіву про серії виробу, наявних на поточний період, дані про компоненти СТО;

- Діагностичні повідомлення. Для цього об'єкта властиві наступні атрибути: дата й час запису повідомлення, номер програми, порядковий номер таблиці БД, типові формулювання повідомлення, логін користувача.

Оскільки повідомлення з однаковим формулюванням може неодноразово входити в таблицю, для зменшення обсягу даних у таблиці й відповідно обсягу пам'яті зайнятого таблицею варто створити відношення, що зберігає код повідомлення й типові його формулювання, і зв'язати журнал повідомлень із переліком повідомлень за допомогою зовнішнього ключа. Таким способом у БД додається наступне відношення: перелік повідомлень.

• Користувачі системи. Для даного відношення властиві наступні атрибути: логін користувача, шифр бригади, ППП користувача, телефон, пароль користувача, схема прав доступу.

Введемо обмеження на складові БД "Параметри моделей СТО", які зберігатимуться у розробленій БЗ. Моделі та множини, які їх складають, детально описані в роботах [29, 167, 168].

Відповідно до виразу (3.2) множина обмежена евклідовим простором $R^{(3)}$ та замкнена. Простір, в якому буде проводитися створення БКС, а, в подальшому, МРО та МПВ, представимо у вигляді деякої області:

$$O = \bigcup_{t=1}^q (R^{(3)} \setminus K_t) \cup O^*,$$

де K_t ($t = 1, 2, \dots, q$; q – кількість областей заборони) – області заборони для розміщення об'єктів в O , O^* – область, геометрична форма та розміри котрої залежать від результату вирішення задачі.

Розглянемо параметри з БД, які використовуються при побудові моделей БКС, та знаходяться в області O через P_i ($i = 1, 2, \dots, n$; n – кількість параметрів).

Множину об'єктів $\{P_i\}_n$ необхідно розмістити всередині області O :

$$\bigcup_{i=1}^n P_i \cap O = \bigcup_{i=1}^n P_i.$$

Об'єднання $\bigcup_{i=1}^n P_i$, яке відображає параметри в БКС, будуються з урахуванням виконання ряду обмежень:

• об'єкти P_i ($i = 1, 2, \dots, n$) мають бути розміщені у межах області O :

$$(P_i \setminus p_i) \cap (R^{(3)} \setminus O) = 0; \quad i = 1, 2, \dots, n,$$

де p_i – межа об'єкту P_i ;

• при наявності областей заборони K_t ($t = 1, 2, \dots, q$) для розміщення P_i всередині області O ($\bigcup_{t=1}^q K_t \subset O$) компоненти БКС не мають з ними пересікатися:

$$(P_i \setminus p_i) \cap (K_t \setminus k_t) = 0; \quad i = 1, 2, \dots, n; \quad t = 1, 2, \dots, q;$$

де k_t – межа об'єкту K_t ;

- компоненти БКС не повинні мати спільних точок перетину:

$$(P_i \setminus p_i) \cap (P_j \setminus p_j) = 0 \quad ; i, j = 1, 2, \dots, n; i \neq j,$$

де $p_{i(j)}$ – межа об'єкту $P_{i(j)}$.

Вищеперераховані обмеження допускають можливість зіткнення компонентів БКС, проте в реальних задачах між компонентами задається відстань (на основі ТВ, ТП, конструктивно-силового набору, стандартів та ін.)

Формулювання мети використання БЗ та БД полягає у визначенні на припустимій множині варіантів рішень, які описуються обмеженнями, такого варіанту, для котрого критерій оптимальності (цільова функція або функціонал), який визначає якість рішення, приймає екстремальне значення: $F^* = \underset{P_n \in E_k(m_U)}{extr} F(P_n)$,

де $E_k(m_U)$ – критерії ефективності параметрів $m_U \in M_U$.

Фактичне вирішення задачі накладення обмежень на БКС, яке представляє конструкцію та, в подальшому, компоувальної схеми, неможливе без визначення вектору параметрів розміщення компонентів БКС $m_U \in M_U$. Відповідно, система виразів, через які позначалися відстані, є обмеженнями для $m_U \in M_U$: $E_k(m_U)$.

Зокрема, нерівності, що входять до системи $E_1(m_U)$ носять геометричний характер. Специфіка побудови СТО накладає також обмеження: $E_2(m_U)$ – центрувальні, обумовленими підсиленням конструкції, і, тим самим, підвищення маси СТО у випадку зміщення центра мас компонентів СТО.

Те саме можна сказати про масу конструкції: $E_3(m_U)$. Навіть якщо відсутні прямі економічні втрати, розробка компонентів СТО може бути неможлива з причини підвищення верхньої межі маси СТО.

Суттєвим є обмеження $E_4(m_U)$ – за зовнішніми впливами на область розробки (різні рівні механічних, кліматичних та інших впливів); сумісність обладнання; вимоги експлуатаційної технологічності та інше.

Обмеження $E_5(m_U)$ – виконання вимог технологічності, що є сполучною ланкою між проектуванням та виробництвом СТО і т.д.

Оператор $E_i(m_U)$ ($i=1\dots k$) розуміється в широкому сенсі. Це може бути рівність чи нерівність, їхня система, а для деяких специфічних обмежень – диференційний або інтегральний оператор. Припустимі рівні $E_i(m_U)$ оговорені рядом нормативних документів, вимогами до створення СТО, ТВ і т. д. ЗВ цілому система обмежень виглядає:

$$\left\{ \begin{array}{l} E_1(m_U)_{\min} \leq E_1(m_U) \leq E_1(m_U)_{\max}; \\ E_2(m_U)_{\min} \leq E_2(m_U) \leq E_2(m_U)_{\max}; \\ \dots\dots\dots \\ E_k(m_U)_{\min} \leq E_k(m_U) \leq E_k(m_U)_{\max}. \end{array} \right. \quad (3.1)$$

В роботі складні фізичні процеси зведемо до більш простих геометричним співвідношенням, до обмежень типу $E_i(m_U)$.

Наведений аналіз задачі з метою отримання аналітичних залежностей, доступних для алгоритмізації і наступних програмних побудов, не усуває всіх перепон на шляху її вирішення.

Застосування апарату класичної математики для вирішення задач оптимізації – варіаційного обчислення – викликає принципові складності навіть за порівняно простих технічних задач. [169] Задача розміщення компонентів моделей СТО є багатоекстремальною. Багатоекстремальність припускає варіанти вирішення, що складаються з етапів локальної та глобальної оптимізації.

При розташуванні компонентів використовуємо метод послідовно-одиначного розміщення [14].

Для цільової функції $\chi = \chi(P_1, P_2, \dots, P_n)$ метод послідовно-одиначного розміщення реалізується у вигляді наступної ітераційної формули:

$$\chi_i(P_{N_1}^*, P_{N_2}^*, \dots, P_{N_i}^*) = \text{extr}_{P_{N_i} \in \Omega^*} \chi(P_{N_1}^*, P_{N_2}^*, \dots, P_{N_{i-1}}^*, P_{N_i}); \quad N_i \in [1, n]; \quad i = 1, 2, \dots, n;$$

де χ_i – часткове значення цільової функції після розміщення i компонентів; $P_{N_\mu}^*$ ($\mu = 1, 2, \dots, i-1$) – вектори параметрів розміщених компонентів; P_{N_i} – вектор параметрів чергового компоненту, що розташовується; O^* – область припустимих значень параметрів; n – кількість компонентів, що розміщуються.

Метод інтерпретується наступним чином. Встановлюється взаємо-однозначна відповідність між компонентами, які розміщуються в області Ω та числами з натурального ряду від 1 до n . Тоді довільна послідовність чисел з n визначає послідовність компонентів, що розміщується.

Нехай $\Pi_\lambda = \{N_1^{(\lambda)}, N_2^{(\lambda)}, \dots, N_n^{(\lambda)}\}$ – деяка послідовність чисел з натурального ряду, де $N_i^{(\lambda)} \neq N_{i+1}^{(\lambda)}$; $i=1,2,\dots,n$; $N_i^{(\lambda)} \in [1,n]$; $\lambda=1,2,\dots,\Lambda$; Λ – кількість ітерацій. Відповідно, i -му члену $N_i^{(\lambda)}$ цієї послідовності відповідає компонент $S_{N_i^{(\lambda)}}$ і навпаки. Якщо задача розглядається стосовно до одного компоненту $S_{N_1^{(\lambda)}}$, тоді цільова функція буде залежати від одного вектору:

$$\chi = \chi(P_{N_1^{(\lambda)}}^*); P_{N_1^{(\lambda)}} \in O^*.$$

Розташуємо компонент $S_{N_1^{(\lambda)}}$ в області O так, щоб вектор його параметрів розміщення $W_{N_1^{(\lambda)}}^*$ задовольняв умові:

$$\chi(P_{N_1^{(\lambda)}}^*) = \text{extr}_{P_{N_1^{(\lambda)}}^* \in \Omega^*} \chi(P_{N_1^{(\lambda)}}), \quad (3.2)$$

що однозначно визначає розташування компонента $m_{N_1^{(\lambda)}}$ в заданій області O . Це розташування вважається фіксованим $P_{N_1^{(\lambda)}}^* = \text{const}$ до закінчення розташування усіх компонентів з $\{m_i\}_n$.

Якщо задача розглядається стосовно до μ компонентів, причому компоненти $m_{N_1^{(\lambda)}}$, $m_{N_2^{(\lambda)}}$, \dots , $m_{N_\mu^{(\lambda)}}$ вже розміщені в попередніх процедурах, тоді цільова функція набуває вигляду:

$$\chi = \chi(P_{N_1^{(\lambda)}}, P_{N_2^{(\lambda)}}, \dots, P_{N_\mu^{(\lambda)}}); P_{N_i^{(\lambda)}} \in O^*; i=1,2,\dots,\mu.$$

Онтологічна схема побудови моделей СТО є основою для побудови бази знань сучасного проектування СТО на етапі МРО. Онтологічна схема повністю описує весь процес проектування даної стадії розробки моделей СТО з урахуванням усіх можливих факторів, які впливають на проектування даного етапу СТО.

Отримана система класів, підкласів, атрибутів і зв'язків реалізована мовою OWL, функції інтерпретації, і правила вибору, реалізовані за допомогою OWL і Java, складають основу БЗ і системи прийняття рішень знання-орієнтованої побудови моделей. Схема бази знань знання-орієнтованої побудови моделей показана на рис. 3.7.

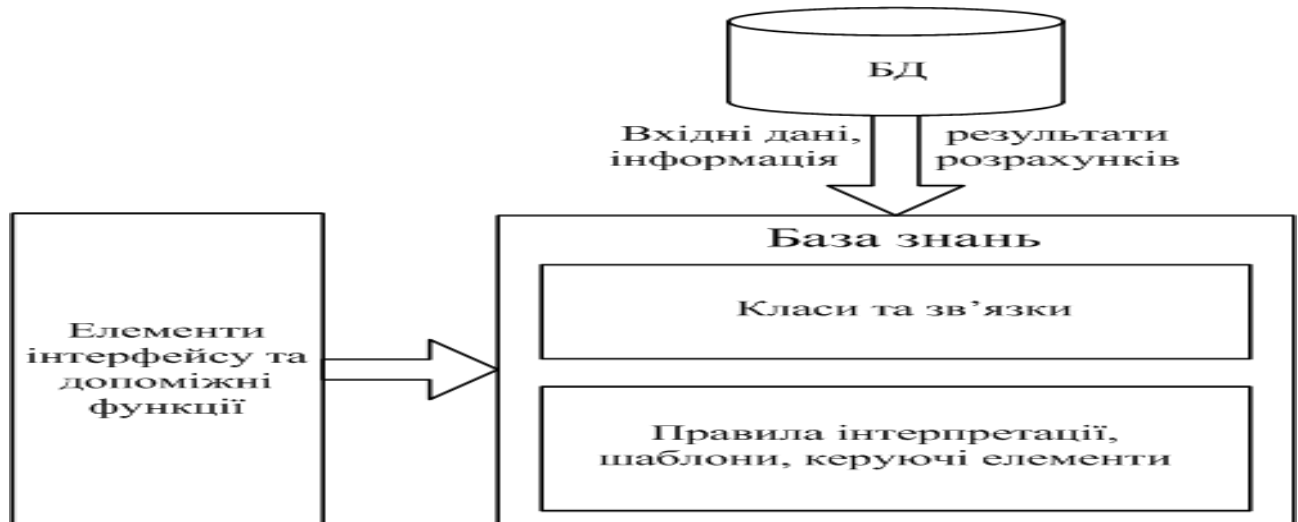


Рисунок 3.7 – Схема бази знань

Розроблені правила в БЗ описують зв'язок між моделями СТО, який визначається системою виводу на основі ЯКЩО-ТОДІ-ІНАКШЕ правил.

Представимо приклад правил:

IF $p_i^{\min} > p_i > p_i^{\max}$ **THEN** p_i – НЕДІЙСНИЙ (стосується обмежень будь-яких моделей);

IF p_{G_i} ЗМІНЮЄТЬСЯ **THEN** ЗМІНЮЮТЬСЯ m_{G_i} , m_{W_i} , m_{A_i} , m_{CC_i} , m_{S_i} , m_{PP_i} , m_{T_i} , m_{E_i} ; і т. д.

Розглянуто задачу побудови залежності параметрів геометричної моделі: m_{G_i} – p_{G_i} від параметрів моделі міцності: m_{S_i} – p_{S_i} та параметрів вагової моделі: m_{W_i} – p_{W_i} компонента СТО. На етапі створення БКС i -го компонента СТО в умовах невизначеності важко оцінити міжмодельний вплив. У зв'язку з чим доцільним є використання для оцінки моделей СТО лінгвістичних змінних (зв'язки σ_{M_U1}). В загальному випадку механізм логічного вислідку включає чотири етапи: введення нечіткості (фазифікація), нечіткий вислід, композиція та приведення до чіткості (дефазифікація). Алгоритми нечіткого вислідку відрізняються головним чином

видом правил, логічних операцій та різновидом методу дефазифікації. Існують моделі Мамдані, Сугено, Ларсена, Цукамото.

В роботі розглянута модель Мамдані, що найчастіше використовується, та містить у консеквентах правил логічного вислуду нечіткі значення. Терм-множину змінної p_{Wi} позначимо: $P_{Wi} = \{P_{Wi}min, P_{Wi}av, P_{Wi}max\}$ та p_{Si} позначимо: $P_{Si} = \{P_{Si}min, P_{Si}av, P_{Si}max\}$. Терм-множину змінної p_{Gi} позначимо: $P_{Gi} = \{P_{Gi}min, P_{Gi}av, P_{Gi}max\}$, де представлено множину значень змінної: $P_{i}min$ – «припустимі до мінімального значення»; $P_{i}av$ – «припустимі», $P_{i}max$ – «припустимі до максимального значення» (на рис. 3.8 зображені функції приналежності термів відповідно до економічно обґрунтованих значень обмежень вагової моделі та значень моделі міцності на прикладі рампи ЛА).

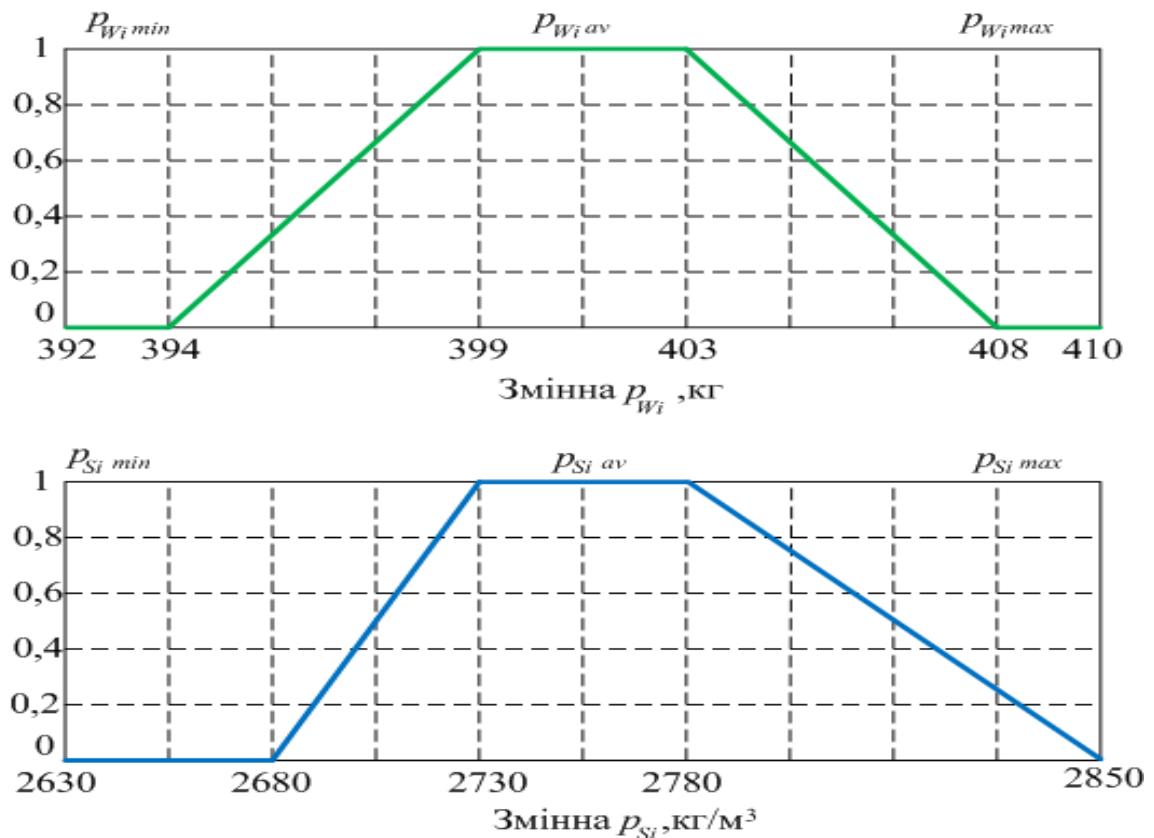


Рисунок 3.8 – Терми вагової моделі та моделі міцності на прикладі рампи ЛА

Наступним етапом побудови нечіткої моделі є створення бази правил:

- 1). IF p_{Wi} IS $P_{Wi}min$ AND p_{Si} IS $P_{Si}min$ THEN p_{Gi} IS $P_{Gi}min$
- 2). IF p_{Wi} IS $P_{Wi}min$ AND p_{Si} IS $P_{Si}av$ THEN p_{Gi} IS $P_{Gi}av$
- 3). IF p_{Wi} IS $P_{Wi}min$ AND p_{Si} IS $P_{Si}max$ THEN p_{Gi} IS $P_{Gi}av$

- 4). IF p_{wi} IS P_{wiav} AND p_{Si} IS $P_{Si\min}$ THEN p_{Gi} IS P_{Giav}
- 5). IF p_{wi} IS P_{wiav} AND p_{Si} IS P_{Siav} THEN p_{Gi} IS P_{Giav}
- 6). IF p_{wi} IS P_{wiav} AND p_{Si} IS $P_{Si\max}$ THEN p_{Gi} IS $P_{Gi\max}$
- 7). IF p_{wi} IS $P_{wi\max}$ AND p_{Si} IS $P_{Si\min}$ THEN p_{Gi} IS P_{Giav}
- 8). IF p_{wi} IS $P_{wi\max}$ AND p_{Si} IS P_{Siav} THEN p_{Gi} IS $P_{Gi\max}$
- 9). IF p_{wi} IS $P_{wi\max}$ AND p_{Si} IS $P_{Si\max}$ THEN p_{Gi} IS $P_{Gi\max}$

У роботі представлено нечіткий вислід Мамдані для задачі побудови залежності геометричної моделі: $m_{Gi} - p_{Gi}$, де показано агрегування нечітких правил при двох вхідних змінних p_{wi} та p_{Si} . Для цього використовується логічний добуток (оператор *min*). Агрегування імплікацій, що стосуються правил, проводиться з використанням логічної суми (оператор *max*).

В моделі Мамдані кожне правило має ступінь свого виконання ω_i , що вираховується наступним чином:

$$\omega_i(p_1 \dots p_{n_p}) = \bigcap_{j=1}^n \mu_{j,i}(p_j), \quad i=1 \dots n_R,$$

де \bigcap – нечітка операція кон'юнкції, що відповідає оператору “AND” в правилах, котра може задаватися за допомогою різних t-норм, як-от: мінімум, добуток та інше; n_p – кількість входів (в даному випадку $n_p = 2$); $\mu_{j,i}(p_j)$ – функція приналежності на j -тому вході в антецеденті i -го правила; n_R – кількість правил (в даному випадку $n_R = 9$).

Після розрахунку ступеню виконання правил, за допомогою імплікації (в системах Мамдані зазвичай використовується операція мінімуму), розраховуються нечіткі значення консеквентів правил. Далі за допомогою операції агрегації (в системах Мамдані зазвичай використовується операція максимуму) розраховуються нечіткі значення виходу з функцією приналежності $\mu_{P_{res}}(p_G)$ у відповідності з виразом

$$\mu_{P_{res}}(p_G) = \bigcup_{i=1}^{n_R} (\omega(p_1 \dots p_{n_p}) \bigcap \mu_{p_{Gi}}(p_G)),$$

де \cup – операція агрегації, що відповідає об'єднанню нечітких правил за ELSE, в системі Мамдані еквівалентно диз'юнкції; \cap – операція імплікації (в системі Мамдані еквівалентна кон'юнкції); $\mu_{p_{G_i}}(p_G)$ – функція приналежності консеквента i -го правила. При використанні максимуму у якості оператора агрегації та мінімуму у якості оператора імплікації процедура отримання нечіткого значення висліду називається *композицією max-min*.

Коли входи правил оброблені описаним вище алгоритмом та отриманий нечіткий вихід $\mu_{p_{res}}(p_G)$, необхідно за допомогою дефазифікації знайти відповідне йому чітке значення p_G^* .

Основними методами дефазифікації є: метод центру ваги, центра сум або усередненого максимуму.

Дефазифікований вислід за методом центру ваги визначається:

$$p_G^* = \frac{\sum_{i=1}^{N_{p_G}} p_{G_i} \mu_{p_{G_{res}}}(p_{G_i})}{\sum_{i=1}^{N_{p_G}} \mu_{p_{G_{res}}}(p_{G_i})},$$

де сумування виконується на дискретних значеннях p_{G_i} області визначення виходу, розділеної на N_{p_G} точок.

За методом центру сум функція приналежності висліду будується шляхом сумування (агрегації за допомогою суми, а не об'єднання по максимуму) вислідів кожного з правил, котрі спрацювали:

$$p_G^* = \frac{\sum_{i=1}^{N_{p_G}} p_{G_i} \sum_{k=1}^n \mu_{0,k}(p_{G_i})}{\sum_{i=1}^{N_{p_G}} \sum_{k=1}^n \mu_{0,k}(p_{G_i})}.$$

При методі дефазифікації необхідно взяти чітке найбільше значення ступеню приналежності функції $\mu_{p_{res}}(p_G)$. У випадку існування декількох елементів області визначення з максимальним значенням ступеню приналежності обирається усереднене значення максимуму:

$$p_G^* = \sum_{m=1}^M \frac{p_{Gm}}{M}.$$

Онтологія початкових етапів побудови моделей СТО була побудована за допомогою засобів Protégé 4.3, її зв'язність та коректність перевірялася за допомогою інструментів даного додатка. Функції доповнення, зміни та перевірки онтології здійснюються за допомогою прикладних програм цієї системи, а також засобів відповідних бібліотек Java, Jena, Pellet. [171 – 173] Онтологічна БЗ відіграє ключову роль у здійсненні процесу знання-орієнтованої побудови моделей за допомогою БКС. Онтологія зв'язує між собою окремі компоненти системи (елементи інтерфейсу, базу даних і прикладні програми), формує необхідні дані для прийняття рішень, керує ходом керування параметрів моделей СТО, створює історію проекту й забезпечує зворотний зв'язок між окремими його станами.

Основна мета онтології створення узагальненої моделі СТО – служити сполучною ланкою між різними компонентами знання-орієнтованої побудови моделей СТО (такими як БД, прикладні програми, інтерфейс, система захисту), а також бути основою для прийняття рішень в процесі знання-орієнтованої побудови моделей СТО.

3.4 Розробка системи програмних модулів зв'язку складових інформаційної технології

В роботі розроблено систему програмних модулів, що створює швидкі та якісні умови для внесення змін в процес проектування і виробництва СТО за допомогою спеціально налаштованих дій програмних модулів та здатне весь час підтримувати дуплексний зв'язок із БД і дозволяє: робити запит та отримувати інформацію про складові, правила, умови та обмеження БЗ.

Головна задача розробки системи програмних модулів розбивається на підзадачі, а її рішення здійснюється як композиційна поведінка програмних модулів, які реалізують вибір і виконання послідовності доступних їм дій, направлених на досягнення їхніх власних цілей. При цьому передбачається

наявність механізмів адаптації програмних модулів, а також алгоритмів їх навчання, частково використовуючи принципи роботи агентів. Агентом є об'єкт, який може сприймати оточуюче середовище за допомогою датчиків і здійснювати на нього вплив, використовуючи маніпулятори. [174] Агентно-орієнтований підхід до програмування — різновид подання програм, або парадигма програмування, у якій основними концепціями є поняття агента і його поведінки, що залежить від середовища, у якому він перебуває. [175]

Робота використовує програмні модулі, БЗ, БД та ІС PLM, щоб реалізувати швидкий та надійний процес знання-орієнтованої побудови моделей СТО.

Структурна схема побудови моделей СТО на основі БД і БЗ показана на рис. 3.9. [176] Зв'язку програмних модулів (в тому числі, агентів), онтології та БЗ присвячено багато робіт в різноманітних галузях [177 – 181], зокрема, при створенні СТО [182, 183]



Рисунок 3.9 – Структурна схема використання програмних модулів для побудови моделей складного технічного об'єкта

Дана структурна схема використовує онтологічну схему побудови моделей СТО, яка представлена у попередньому підпункті, як базу знань побудови моделей СТО на етапі МРО.

Зокрема, схему побудови моделей СТО можна в загальному описати наступним чином:

База знань початкових етапів побудови моделей СТО зберігає в собі основні складові та правила, умови і обмеження процесу побудови моделей СТО етапу МРО, яку названо «Створення моделей СТО».

Система програмних модулів за допомогою спеціально налаштованих дій програм здатне весь час підтримувати прямий та зворотній зв'язок із базою знань: робити запит та отримувати інформацію про складові, правила, умови та обмеження бази знань.

Налаштовані модулі акцентують увагу на зміні даних в БД "Параметри моделей СТО", вносячи необхідні зміни в параметри моделей різноманітних частин СТО, відповідно вимогам БЗ.

Функціональна схема використання програмних модулів для побудови моделей СТО показана на рис. 3.10.

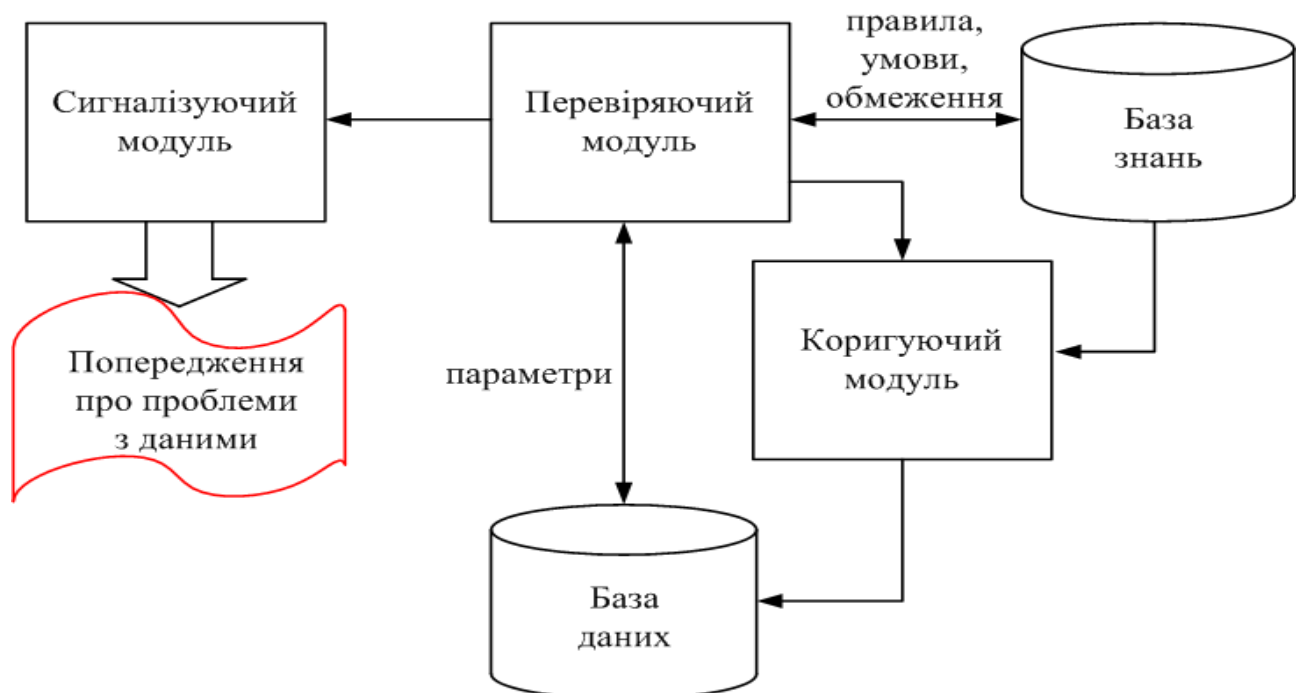


Рисунок 3.10 – Функціональна схема використання програмних модулів для побудови моделей складного технічного об'єкта

Функціональна схема докладніше описує принцип роботи програмних модулів з базою знань та базою даних побудови моделей СТО.

Алгоритм роботи функціональної схеми побудови моделей СТО за допомогою застосування програмних модулів та БЗ більш детально можна описати наступним чином:

Представлена система програмних модулів для побудови моделей СТО на етапі розробки МРО для зв'язку БЗ та БД складається з трьох програмних модулів: Перевіряючого, Коригуючого та Сигналізуючого.

Перевіряючий модуль надсилає запит до необхідної комірки БД "Параметри моделей СТО", в котрій зберігається інформація про параметри, які надалі використовуються для побудови БКС та моделей. Далі значення з даної комірки перевіряються програмним модулем на відповідність значень вимогам БЗ.

Коригуючий модуль на основі закладених правил та описаної логіки приймає рішення про внесення зміни одних параметрів відносно значень інших та відправляє значення параметрів в БД.

Результат перевірки Перевіряючий модуль надсилає Сигналізуючому модулю. Якщо результат перевірки показує відповідність даних параметрів БД вимогам БЗ, тоді виводиться повідомлення про можливість використання даних. Якщо результат перевірки показує, що дані параметрів БД не відповідають вимогам БЗ, тоді виводиться повідомлення з попередженням про проблеми з даними. В цьому випадку треба буде змінювати параметри в БД відповідно обмеженням та вимогам БЗ.

Зв'язок знання-орієнтованого керування параметрами моделей відбувається за допомогою середовища Java, в якому програмні модулі використовують інформацію про параметри моделей з бази знань відповідно до функціональної схеми використання програмних модулів для створення моделей СТО на рис. 3.10.

Розподілені об'єктні архітектури, маючи багато переваг, мають наступні недоліки: необхідність перекомпіляції програмних кодів при внесенні змін в об'єкти й інтерфейси; неможливість динамічної адаптації поведінки програмних

об'єктів залежно від станів і поводження середовища; неможливість роботи в явній формі з моделями знань.

Обране в роботі розподілене об'єктне середовище, засноване на Java VM, надає програмним модулям наступні властивості:

- Автономність (здатність виконувати дії самостійно: програмному модулю треба список поведінки: працювати у режимі очікування, в режимі «зайнятий – іде робота»).
- Стійкість (не маючи побічного ефекту після кожного поводження, має можливість відкочування).
- Активне поводження (у випадку невдачі, виконуються циклічні спроби виконати поточне, а потім перейти до наступної дії поводження. Під час виконання поводження програмний модуль зайнятий – не взаємодіє із середовищем. Після завершення поточного поводження, програмний модуль переходить до наступного за списком, а у випадку відсутності списку – очікує появу такого).
- Мобільність: можливість передавати програмний модуль в інший контейнер. Дана властивість виникає з відсутності після виконання кожного поводження побічного ефекту.

3.5 Висновки

1. Розроблено модифікацію методу для керування параметричною інформацією при побудові моделей СТО; обміну даними між складовими ІТ СALS та PLM-рішень, а також засоби реалізації методу, що є базою для розробки інформаційної технології та виявлені недоліки існуючих методів.

2. Розглянуто питання формування засобів забезпечення реалізації інформаційної технології створення та супроводження узагальненої моделі СТО, які розширюють функціональні можливості процесу побудови моделей СТО відповідно до виділених завдань проблемної орієнтації. Розроблено базу даних побудови моделей СТО.

3. Представлено онтологічну схему розробки узагальненої моделі СТО, що забезпечує високу якість створення складного технічного об'єкта; раціоналізацію прийняття конструкторських рішень при ув'язці компонентів конструкції, систем і обладнання СТО; зниження строків створення СТО при формуванні віртуальної моделі замість створювання натурального макета; зниження матеріальних і трудових витрат; підвищення ергономічних показників роботи конструктора-проектувальника.

4. Розроблено базу знань для керування та обробки інформації про основні параметри та обмеження в моделях СТО, що дозволяє представити дані процесу проектування у вигляді впорядкованої структури з чітко визначеними зв'язками.

5. Розроблено та впроваджено систему програмних модулів в процес розробки знання-орієнтованої побудови моделей СТО, що створює швидкі та якісні умови для внесення змін в процес проектування та виробництва СТО.

4 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ СТВОРЕННЯ ТА СУПРОВОДЖЕННЯ УЗАГАЛЬНЕНОЇ МОДЕЛІ СКЛАДНИХ ТЕХНІЧНИХ ОБ'ЄКТІВ

Вирішення задачі реалізації інформаційної технології створення та супроводження узагальненої моделі СТО засновано на використанні запропонованих принципів та розробленого методу знання-орієнтованого створення СТО.

Реалізація інформаційної технології створення та супроводження узагальненої моделі СТО із використанням знання-орієнтованих технологій відбувається, виходячи з вимог раціонального проектування та виробництва складних технічних об'єктів, використовуючи: ПЗ для реалізації бази знань, бази даних, написання модулів; складові ІТ CALS та PLM-рішень.

4.1. Структурна схема інформаційної системи створення та супроводження узагальненої моделі складних технічних об'єктів

На теперішній час застосування знання-орієнтованих технологій при створенні СТО ще дуже нова і незвична галузь. Правила і методики, що використовуються при побудові цих технологій ще не досить визначені, і, хоча існує багато онтологій різних предметних областей, їх застосування не розповсюджене і розглядається більше як наукове дослідження.

На основі виконаного аналізу і теоретичних підходів запропоновано структурну схему ІТ створення та супроводження узагальненої моделі СТО (опис зв'язків σ_{MU3}), що зменшує час створення моделей за рахунок відділення параметрів від моделей та дозволяє керувати однією узагальненою моделлю СТО протягом її життєвого циклу з різних складових ІТ CALS та PLM-рішень. Впровадження запропонованого математичного опису призводить до оптимізації процесу створення моделей СТО через введення в процес створення узагальненої

моделі можливості керування моделями СТО з різних ІС, які містяться в проекті СТО в ІС PLM (рис. 4.1).

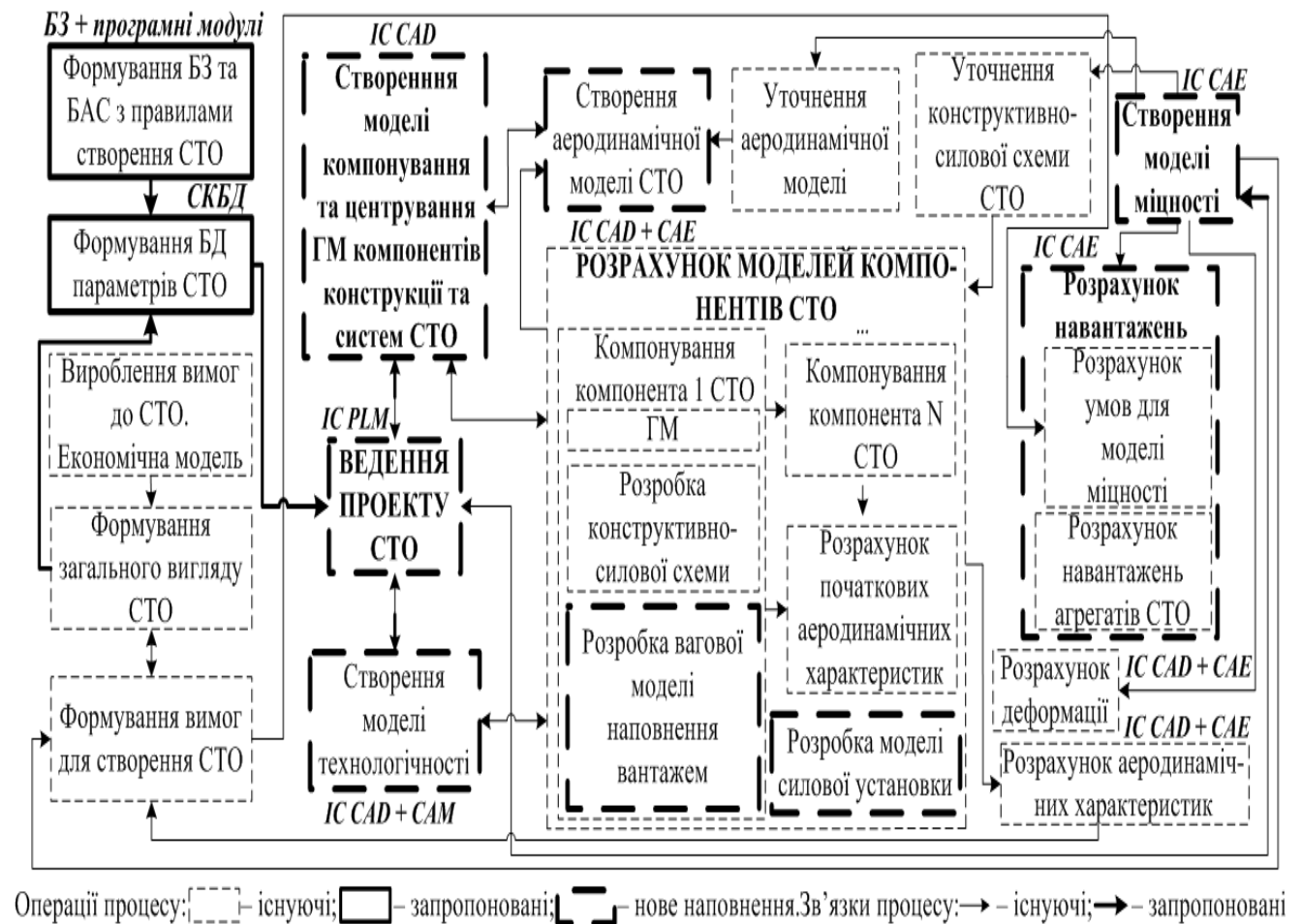


Рисунок 4.1 – Структурна схема інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів

Головними вимогами, якими керуються при застосуванні знання-орієнтованих технологій при створенні моделей СТО, є розробка точного, повного, правильного та керованого процесу побудови моделей СТО.

Етапи по впровадженню ЗОТ при створенні моделей СТО:

1. Опис процесу впровадження ЗОТ при створенні моделей компонента конструкції, систем та обладнання СТО.
2. Визначення параметрів для керування параметричною інформацією моделей компонента конструкції, систем та обладнання СТО;

3. Використання створеного компонента конструкції, систем та обладнання моделей СТО для обміну даними в необхідних складових IT CALS та PLM-рішень.

Виконання вказаних вище дій забезпечується за допомогою введення до складу впровадження ЗОТ моделей СТО механізму прийняття рішень та онтологічної бази знань.

Структурна схема ІС створення та супроводження узагальненої моделі СТО на підприємстві показана на рис. 4.2, де представлено приєднання розроблених інструментальних засобів ІТ до існуючих серверів підприємства, з яких отримуються дані.

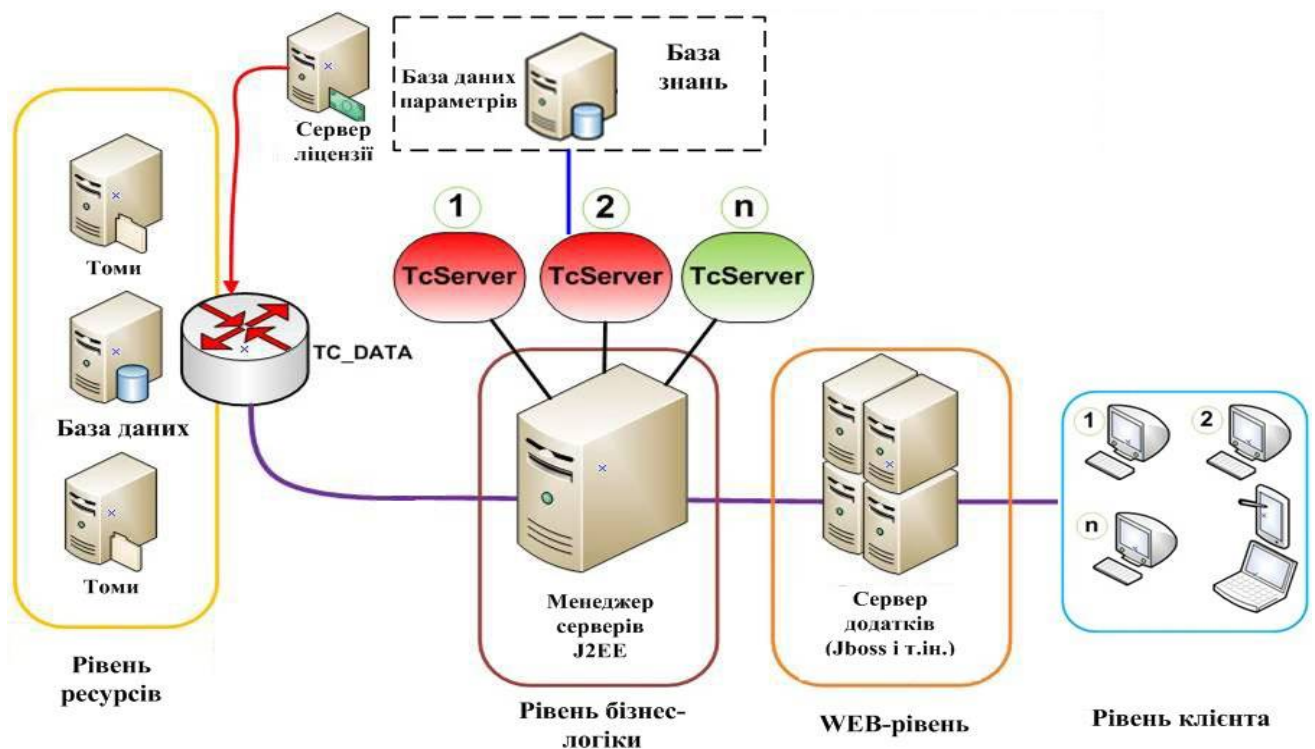


Рисунок 4.2 – Структурна схема інформаційної системи створення та супроводження узагальненої моделі складних технічних об’єктів

Основними компонентами ІС створення та супроводження узагальненої моделі СТО є:

1. Модуль роботи з БЗ, БД та прийняття рішень: основний модуль системи, що організує взаємодію усіх її частин, забезпечує проведення процесу проектування, яке базується на даних, отриманих з БЗ.

2. Модуль роботи з ІС САх: організує та планує ітерації та етапи проведення побудови моделей шляхом організації зв'язку з ІС САх та, в результаті, з ІС PLM.

3. Модуль захисту інформації: забезпечує ідентифікацію та користувачів, організує різні види доступу до ведення процесу впровадження ЗОТ при створенні моделей СТО.

Основні компоненти ІС створення та супроводження узагальненої моделі СТО показано на рис. 4.3.

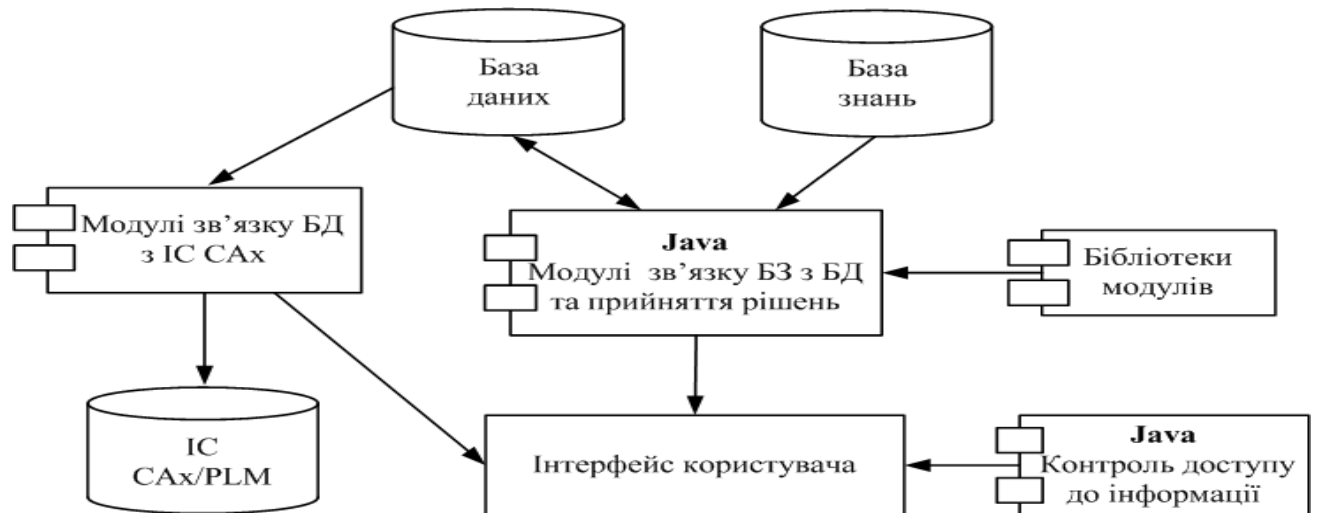


Рисунок 4.3 – Основні компоненти інформаційної системи створення та супроводження узагальненої моделі складних технічних об'єктів

База знань, яка лежить в основі системи прийняття рішень знання-орієнтованої побудови моделей СТО, взаємодіє в процесі роботи з кожним з основних компонентів знання-орієнтованої побудови моделей СТО:

- з модулем роботи з БД, БЗ та прийняття рішень на всіх етапах процесу побудови моделей СТО, визначенні наступного етапу побудови моделей та інших моделей СТО, зміні та доповненні БД та БЗ;
- з модулем роботи з САх при наповненні, зміні та керуванні вхідних параметрів моделей СТО;
- з базою даних моделей СТО в PLM системі для забезпечення передачі інформації та правильного вибору даних;
- з файловою БД при проведенні усіх операцій з параметрами моделей СТО, пов'язуючи її з іншими частинами програми;

- з інтерфейсом користувача для організації вибору інформації, перевірки даних, введених користувачем;
- з модулем захисту інформації для забезпечення надійного захисту даних.

Під онтолого-керуючою архітектурою розуміється така архітектура системи, в якій онтологія предметної області спільно з іншими компонентами формує механізм реалізації конкретного додатку, наприклад, онтолого-керуючої системи інформаційного пошуку, онтолого-керуючого програмного комплексу, онтолого-керуючого електронного навчання та ін. [184 – 190] При цьому онтологія предметної області описана певною мовою опису (список мов визначений, зокрема, консорціумом W3C, наприклад: OWL, RDFS та ін.) Розглядаючи ступінь впливу онтології на складові впровадження ЗОТ при створенні моделей СТО, враховуючи можливість онтології змінюватися з часом, можна говорити про онтолого-керуючу інформаційну систему. [191]

4.2 Реалізація бази даних та бази знань створення узагальненої моделі складних технічних об'єктів

Проблемою керування змінами параметрів наразі ще не займаються в повній мірі провідні постачальники програмного забезпечення IC САХ та PLM систем. У той же час, керування великою кількістю взаємопов'язаних параметрів призводить до стану, коли зміна в одній з параметричних моделей впливає на велику кількість моделей.

В роботі запропоновано використання онтології як складової БЗ для представлення параметрів моделей та їх взаємозв'язків з тим, щоб полегшити процес перевірки змін і попереднього виявлення змін в параметричних моделях.

Наприклад, в складальній одиниці з 10 середньої складності моделей містяться понад 4200 параметрів (з них близько 150 користувальницьких). [192] А тисячі моделей мають мільйони параметрів і створюють проблеми керуванню параметрами моделей.

Тому розроблено інструмент, який допоможе проектувальникам і обслуговуючому персоналу в процесі керування параметрами моделей.

Онтологічні методи представляються як вирішення цієї проблеми. Онтологія є знанням про параметри та їх взаємозв'язки. Додатково має бути розроблено забезпечення прозорі взаємодії з ІС САХ, PLM та БЗ.

Алгоритм використання ЗОТ при розробці моделей СТО:

- Накласти обмеження та умови в БЗ на параметри, які містяться в БД відповідно до ТВ та ТП.
- Перевірити відповідність параметрів в БД умовам та обмеженням з БЗ за допомогою програмних модулів.
- Дозволити керування моделями ззовні ІС САХ за допомогою модулів зв'язку БД та ІС САХ.
- Представити інформацію моделей з ІС САХ в ІС PLM.

Потоки даних впровадження ІТ створення та супроводження узагальненої моделі СТО (рис. 4.4), де представлено результати знання-орієнтованого створення моделей, вдосконалюючи роботу підприємстві, при цьому не руйнуючи усталену систему зв'язків при узгодженні спільної роботи зі створення СТО.

Розроблені моделі СТО на етапі ЕП та РП є даними, які використовуються для проведення вагових розрахунків автоматизованою системою вагового проектування та контролю (АСВПК). Впровадження ЗОТ при створенні моделей СТО є результатом для передачі наступних потоків інформації: розробка БКС; надання попередніх характеристик з міцності СТО (користується відділ міцності); надання попередніх характеристик з технологічності СТО (користується відділ технологічності); видача звітів про поточний стан проекту СТО керівництву; побудова експериментальних моделей, які використовуються як моделі для зв'язку між різними моделями СТО; керування параметрами при розробці моделей на етапі розробки МРО; видача профільним конструкторським відділам (КВ) СТО основних параметрів керування частинами моделей СТО на етапі розробки МПВ.

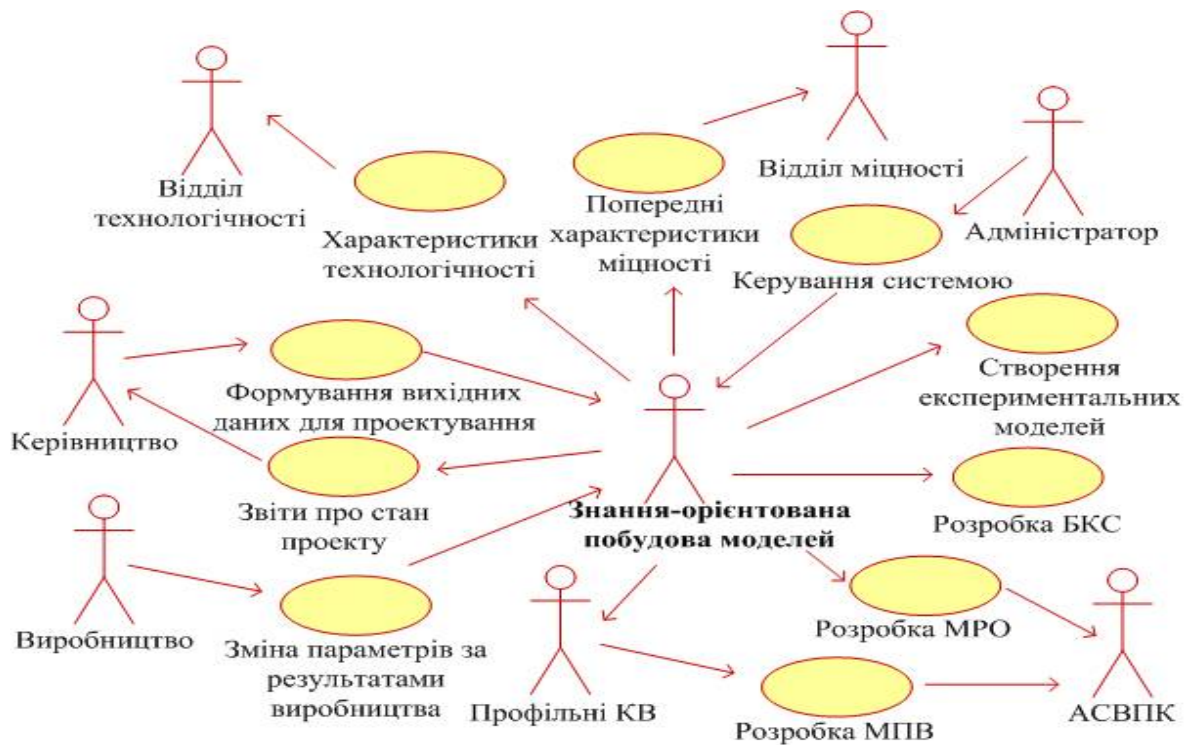


Рисунок 4.4 – Потоки даних впровадження інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів

На рис. 4.5 представлено розробку ІС створення та супроводження узагальненої моделі СТО на прикладі підприємства авіаційної галузі за рахунок використання моделі технологічності та поєднання її з моделями СТО, отриманих з різних ІС.

На рис 4.6 представлено основні зв'язки між складовими компонентами знання-орієнтованої побудови УМ створення складного технічного об'єкта на прикладі найбільш розповсюджених ІС САх та PLM.

Розглянемо розроблену файлову базу даних "Параметри моделей СТО".

Для цього потрібно розглянути детально всі отримані відносини параметрів моделей СТО. Для різних СТО в БД повинні зберігатися найрізноманітніші параметри, що описують його характеристики, особливості констрування, економічні параметри і т.д.

Для прикладу ЛА: параметри можна розбити на наступні групи: літнотехнічні характеристики (ЛТХ); економічні параметри; маса; параметри шасі; параметри силової установки (СУ); параметри вертикального оперення (ВО);

горизонтального оперення (ГО); параметри крила; параметри фюзеляжу; параметри вантажного люку (для транспортних ЛА).

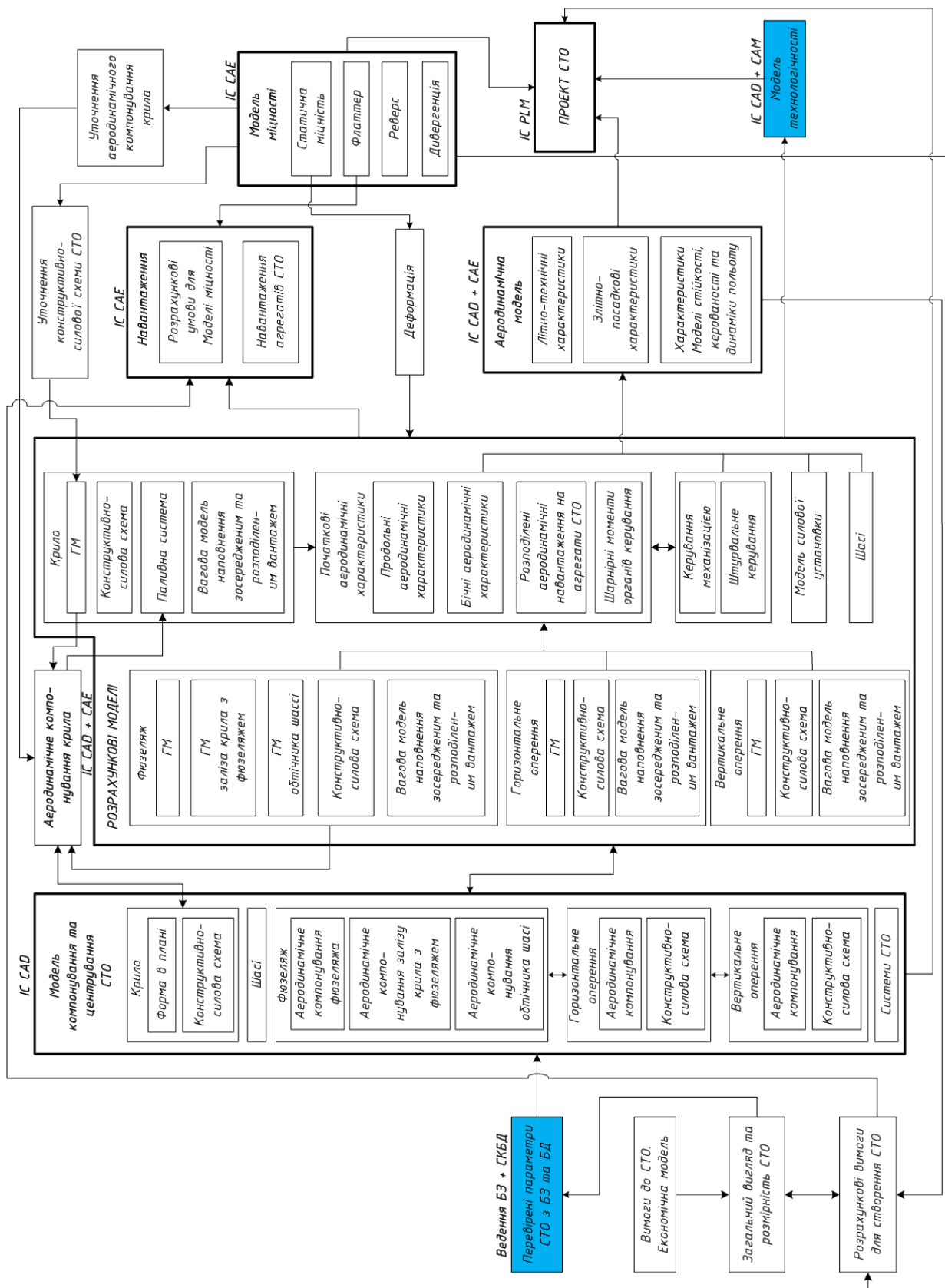


Рисунок 4.5 – Інформаційна система створення та супроводження узагальненої моделі СТО на прикладі підприємства авіаційної галузі

В табл. 4.1 наведено список основних параметрів в БД "Параметри моделей СТО", а у додатку А ці основні параметри в БД "Параметри моделей СТО" розкриті та розшифровані.

Файлова база даних ведеться за допомогою СКБД DbVisualizer [193, 194] – особливо багатого, інтуїтивного інструменту ведення багатьох баз даних для розробників та адміністраторів БД, забезпечуючи єдиний потужний інтерфейс через широке різноманіття операційних систем. З його зручним в роботі та "чистим" інтерфейсом, DbVisualizer, визначається одним з найбільш ефективних в затратах серед доступних інструментів ведення баз даних, що підтримує головні доступні операційні системи. Дані в БД не є остаточними. Параметри в БД можуть доповнюватися або переглядатися відповідно до вимог, які висуваються до типу та характеристик СТО, який розроблюється. Перелік БД "Параметри моделей СТО", які ведуться в СКБД, показано на рис. 4.7, на рис. 4.8 параметри моделей СТО, які зберігаються в БД.

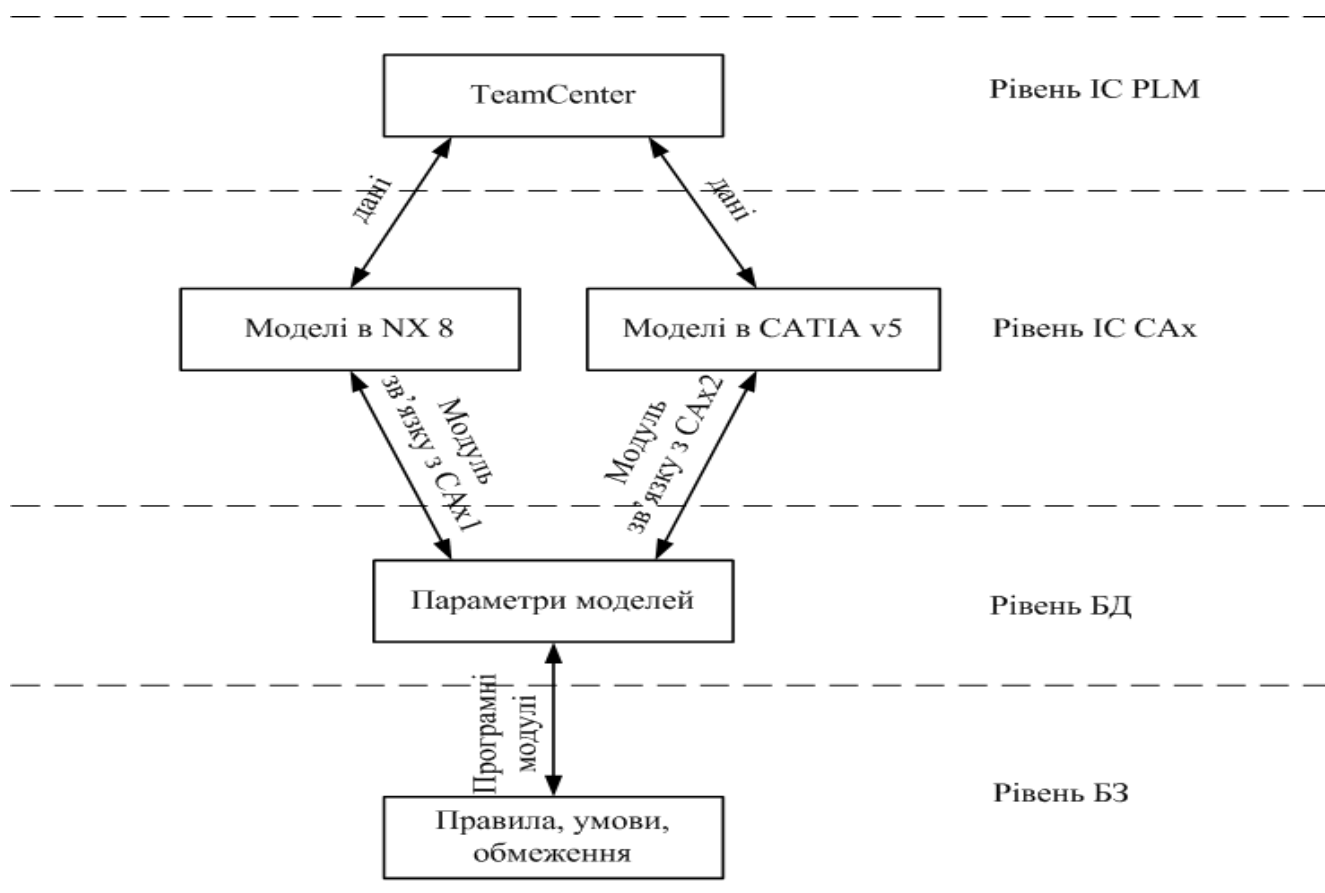


Рисунок 4.6 – Основні зв'язки між складовими компонентами знання-орієнтованої побудови моделей складного технічного об'єкта

Таблиця 4.1 Список співвідношень в базі даних
"Параметри моделей літальних апаратів"

Назва	Співвідношення, що містить
DB_APCH	ЛТХ
DB_ECON	економічні характеристики
DB_MASS	маса
DB_LG	параметри шасі
DB_PP	параметри СУ
DB_TP	параметри ГО
DB_VT	параметри ВО
DB_W	параметри крила
DB_FUS	параметри фюзеляжу
DB_CD	параметри вантажного люку

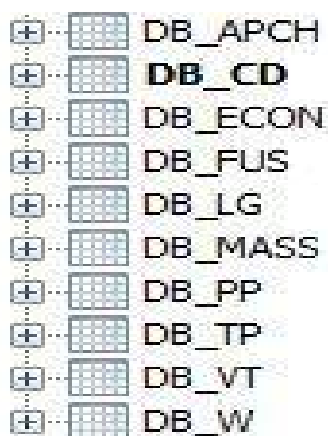


Рисунок 4.7–Бази даних "Параметри моделей літального апарату"

Онтологічна БЗ створення моделей СТО на прикладі ЛА спочатку перевірялася в пакеті Protégé 4.3, в якому є можливість повністю перевірити усі зв'язки всередині онтологічної схеми. У додатку Б представлено вікно програмного продукту Protege 4.3, в якому показано створену БЗ моделей СТО на прикладі ЛА.

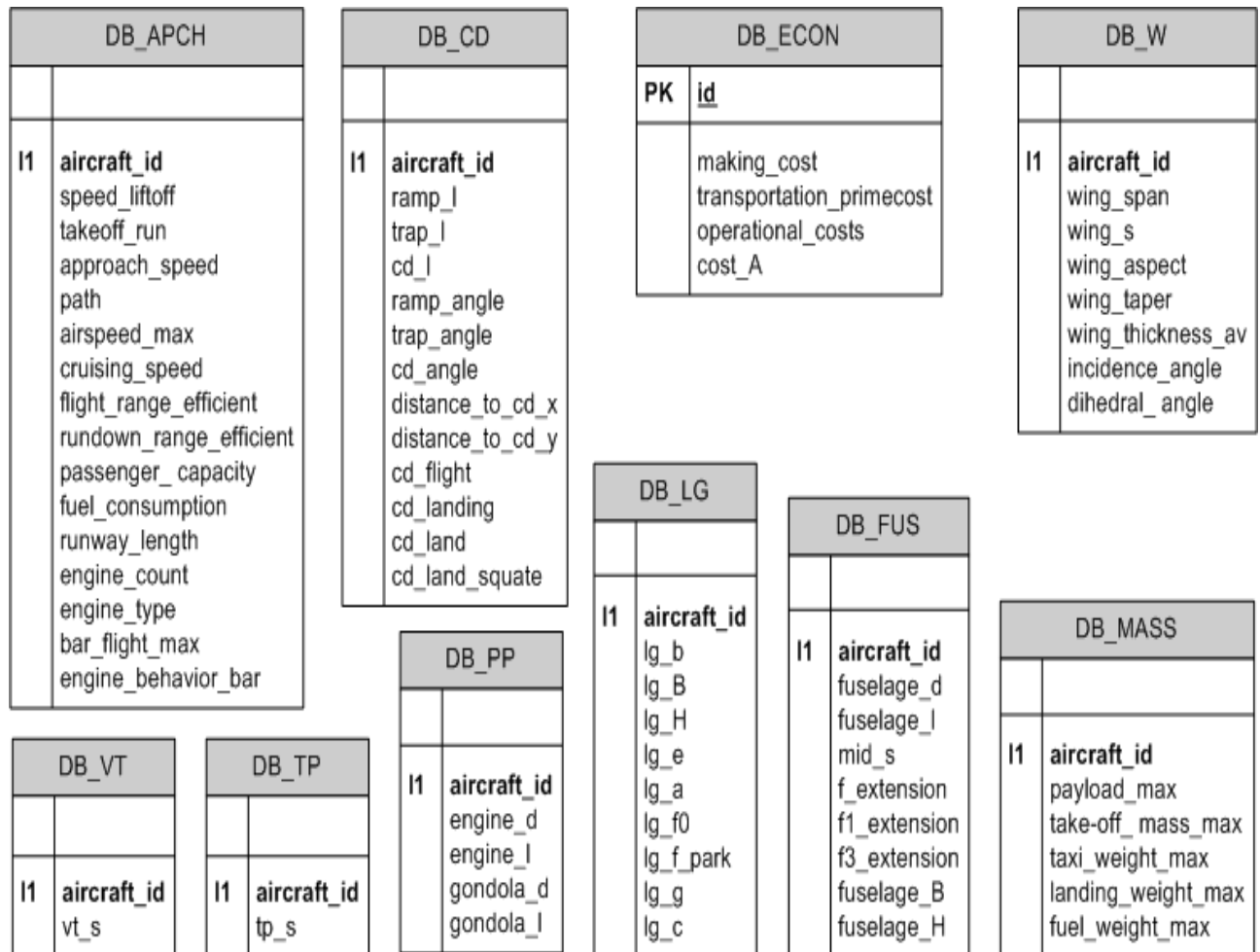


Рисунок 4.8 – Параметри моделей літальних апаратів

На рис. 4.9 представлено граф зв'язку між усіма описаними класами та підкласами онтологічної схеми проектування СТО через введення властивостей (Properties), які «зв'язують» класи та підкласи між собою. Наприклад, згідно онтологічної схеми, реалізовано: властивість «used» («використовується») зв'язує «Рекомендації проектування» та «Найменування моделі» з текстового файлу атрибутивної інформації. Програмний код RDF/XML за результатами моделювання в пакеті Protégé 4.3 повністю представлений у додатку Б.

Програмний код описує усі підкласи онтологічної схеми створення моделей СТО та доводить, що одні класи не можуть порушувати роботу інших класів.

На рис. 4.10 представлено результат моделювання онтологічної схеми проектування СТО, яка відповідає описаному методу застосування онтології при створенні моделей СТО з чіткою перевіркою усіх внутрішніх зв'язків між класами та підкласами в БЗ моделей СТО, використовуючи додаток OWLViz пакету Protege 4.3.

На рис. 4.11 представлено перелік таблиць бази знань, які ведуться у програмі DbVisualizer, де знаходяться обмеження на параметри моделей СТО.

Впровадження БЗ у процес створення моделей СТО забезпечує [195 – 197]:

- Комплексний підхід до керування параметрами на різних етапах створення моделей.
- Використання одних параметрів моделей в подальшому в різних системах САх.

В роботі представлено реалізацію методу моделювання на основі знання-орієнтованих технологій на прикладі вантажного люку (ВЛ) ЛА. На рис. 4.12 наведено БЗ параметрів моделей на прикладі вантажного люку ЛА.

Для представлення знань в БЗ використаємо описану у розділі 3 онтологічну модель. Онтологію певної галузі знань разом з відомостями про властивості конкретних об'єктів можна назвати БЗ. Спочатку онтологія була побудована за допомогою пакету Protege 4.3, її зв'язність і коректність перевірялася за допомогою інструментів даного додатку. Деякі рисунки у роботі були зроблені за допомогою OntoViz – модуля Protege-4.3. Після закінчення роботи над реалізацією ІТ, розробка БЗ відбувається за допомогою СКБД DbVisualizer з підключенням Oracle, а також засобів відповідних бібліотек Java.

Рампові ЛА Ан-124, Ан-225, а також новітні Ан-178, Ан-132, приносять найбільші доходи в українській авіавантажній сфері та є надійними в обслуговуванні потреб замовників. [198]

На основі вищеперерахованих переваг даного типу ЛА в роботі представлено приклад керування вантажного люку ЛА. БД в роботі ведеться за допомогою

СКБД DbVisualizer з підключенням Oracle. На рис. 4.13 представлено фрагмент БД "Параметри моделей ЛА" із СКБД DbVisualizer, що стосується саме ВЛ.

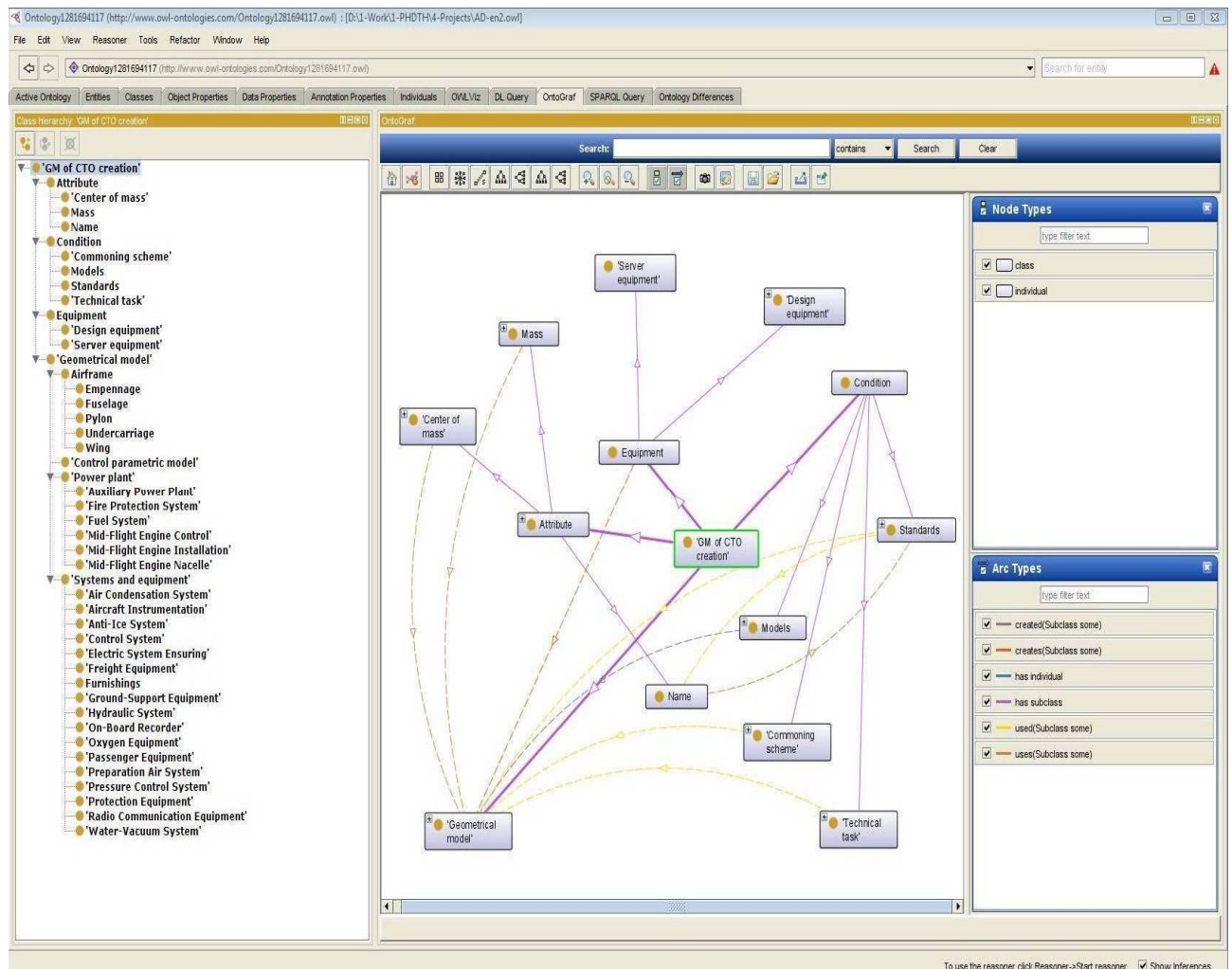


Рисунок 4.9 – Граф зв'язку класів в базі знань моделей в Protege

Зокрема, в БД "Параметри моделей ЛА" представлено наступні параметри ВЛ відповідно до ТВ та ТП у чотирьох положеннях:

- довжина рампи: $RAMP_L = 3420$ мм;
- довжина трапу: $TRAP_L = 2040$ мм;
- довжина ступки: $CD_L = 6850$ мм;
- відстань до осі обертання ступки по X: $DISTANCE_TO_CD_X = 9760$ мм;
- відстань до осі обертання ступки по Y: $DISTANCE_TO_CD_Y = 3210$ мм;
- кути розташування складових ВЛ.

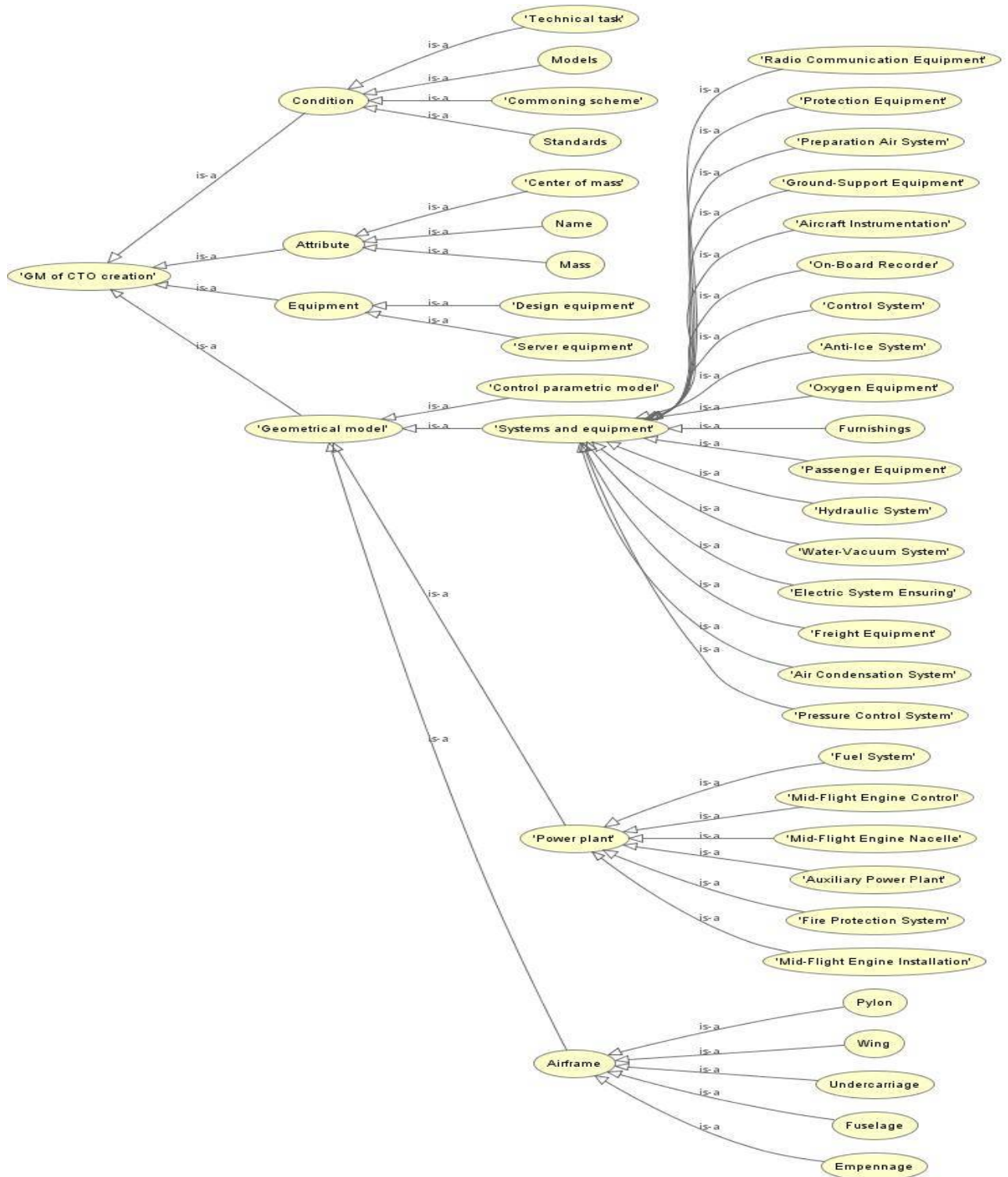


Рисунок 4.10 – Схема розробленої онтології в OWLViz

		KB_CD
		KB_FUS
		KB_LG
		KB_PP
		KB_TP
		KB_VT
		KB_W

Рисунок 4.11 – Перелік таблиць бази знань обмежень параметрів моделей

id	mass_grade	mass
1	0	392
2	0	393
3	0	394
4	0.2	395
5	0.4	396
6	0.7	397
7	0.8	398
8	1	399
9	1	401
10	1	402
11	1	403
12	0.8	404
13	0.6	405
14	0.3	406
15	0.2	407
16	0	408
17	0	409
18	0	410

id	density_grade	density	mass_id	density_id
1	0	2630	2	1
2	0	2680	4	2
3	0.4	2700	5	3
4	0.7	2710	6	4
5	1	2730	7	5
6	1	2740	9	6
7	1	2750	10	7
8	1	2760	12	8
9	1	2780	13	9
10	0.8	2800	14	10
11	0.3	2840	15	11
12	0	2850	17	12

variant	name	ramp_l	ramp_angle	trap_l	trap_angle	cd_l	cd_angle	distance_to_cd_x	distance_to_cd_y
1	max	3453	20	2100	91	6900	0	9800	3300
2	min	3351	-15	2000	-5	6800	-20	9700	3200

Рисунок 4.12 – Приклад бази знань параметрів моделей

VARIANT	NAME	RAMP_L	RAMP_ANGLE	TRAP_L	TRAP_ANGLE	CD_L	CD_ANGLE	DISTANCE_TO_CD_X	DISTANCE_TO_CD_Y
1	cd_flight	3420	16.87	2040	90	6850	-19.2	9760	3210
2	cd_landing	3420	0	2040	0	6850	0	9760	3210
3	cd_land_squate	3420	-9.77	2040	7.4	6850	0	9760	3210
4	cd_land	3420	-14	2040	-2.3	6850	0	9760	3210

Рисунок 4.13 – База даних "Параметри моделей літального апарату"

У таблиці 4.2 представлено положення компонентів вантажного люку ЛА.

Таблиця 4.2. Положення компонентів вантажного люку літального апарату

Положення	Кут розташува ння рампи (градуси)	Кут розташува ння трапу (градуси)	Кут розташува ння стулки (градуси)	Відстань до трапу горизонта льна (мм)	Відстань до трапу вертикаль на (мм)
Положення ВЛ під час польоту	16,87	90	19,2	0	0
Положення ВЛ при десантуванні війська та вантажів	-18,51	18,51	0	3100	2900
Положення ВЛ при розташуванні ЛА на землі	-9,77	-2,23	0	0	0
Положення ВЛ при розташуванні ЛА на землі "з присіданням"	-14	-2,23	0	0	0

4.3 Реалізація системи програмних модулів зв'язку складових інформаційної технології

Задача розміщення трьох основних компонентів вантажного люку: рампи, ступки та трапу ($m_{N_1^{(\lambda)}}, m_{N_2^{(\lambda)}}$ та $m_{N_3^{(\lambda)}}$ відповідно) відбувається наступним чином: Після розміщення компонента $m_{N_1^{(\lambda)}}$ так, щоб вектор його параметрів розміщення задовольняв умові (3.2), тоді другому компоненту відповідає компонент $m_{N_2^{(\lambda)}}$, а цільова функція має вигляд:

$$\chi = \chi(P_{N_1^{(\lambda)}}, P_{N_2^{(\lambda)}}); P_{N_i^{(\lambda)}} \in \Omega^*; i = 1, 2.$$

Розглянемо систему прийняття рішень, що реалізовано за допомогою системи програмних модулів, на прикладі створення моделей ВЛ ЛА. При проектуванні ВЛ ЛА визначальним елементом, на який відбувається максимальне навантаження при завантаженні та вивантаженні, є рампа. Вибір довжини рампи визначається з наступних умов:

- вимог відділення аеродинаміки із забезпеченням безпечної стійкості ЛА при скиданні максимальної маси вантажу з торця рампи при швидкості десантування;
- забезпечення необхідного кута наїзду при завантаженні колісної або гусеничної техніки.

На літаках розробки до 1986 р. забезпечувалася вимога по куту наїзду не більше 15° . Після 1986 р. згідно вимог ОТТ ВВС-86 кут наїзду має бути не більше 12° , а згідно європейських штабних вимог (FTA) не більше 10° . Як правило, довжина рампи визначається з умов її спирання на землю з забезпеченням необхідного кута наїзду та прийнятної будівної висоти зовнішньої жорсткої опори (не більше 80мм). Отримана такими операціями довжина рампи, надалі узгоджується з аеродинаміками, що скидання з такої рампи прийнятне для ЛА. Трап є допоміжним елементом при завантаженні та вивантаженні. Розглянемо процес проектування складової ВЛ – ступкового відсіку (ступки). Ступка закриває проріз ВЛ за рампою. Як правило, ступку (якщо вона одна) або ступки (якщо їх 2

або 3) шарнірно навішують на фюзеляж. Класична схема стулкового відсіку – коли стулка одна, та вона герметична. Родоначальником такої схеми є АН-22 «АНТЕЙ». На АН-132,178 стулка одна, шарнірно навішена на хвостову частину фюзеляжу. Літаки С-17А, А400М, С-295, С-27 – мають одну повністю герметичну стулку.

При проектуванні літаків АН-132, АН-178 розробляються наступні положення розташування компонентів вантажного люку ЛА:

- Положення ВЛ під час польоту;
- Положення ВЛ при десантуванні війська та вантажів;
- Положення ВЛ при розташуванні ЛА на землі;
- Положення ВЛ при розташуванні ЛА на землі "з присіданням".

В роботі створена система програмних модулів (пакет Agent) в середовищі Java, не використовуючи розподілені об'єктні архітектури (CORBA, DCOM, Java RMI, WEB-services, JADE [199]), недоліки яких описані в розділі 3.

Система програмних модулів є системою прийняття рішень в процесі вирішення задачі побудови моделей вантажного люку ЛА. Система включає наступні модулі: Перевіряючий, Корируючий та Сигналізуючий. Фактична робота, що програмний модуль повинен робити, як правило, здійснюється в рамках «поводження модуля». Поводження являє собою завдання, що програмний модуль має виконувати.

В Сигналізуючому програмному модулі (Diagnostic) описано створення загальної форми, яка використовуються іншими модулями для виведення інформації в залежності від заданих умов:

```
public Diagnostic(String frameName){
    prepareGUI(frameName);}
```

Створено Перевіряючий програмний модуль (CheckInput), який перевіряє тотожність введених в БД параметрів на відповідність обмежень в БЗ, визначивши метод його ініціалізації. У випадку, коли параметри в БД не відповідають БЗ, Перевіряючий програмний модуль після порівняння виводить повідомлення "Довжина рампи не відповідає встановленим обмеженням":

```
public class CheckInput {
```

```

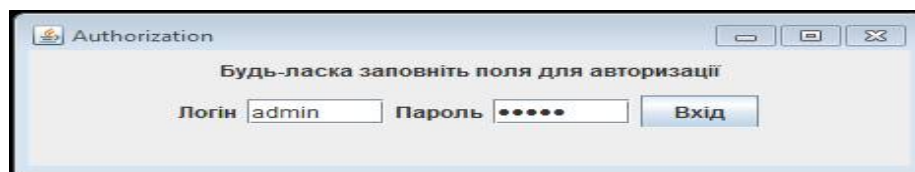
public boolean passFlag;
public CheckInput(/*Diagnostic diagnostic,*/String RAMP_L,
String RAMP_angle, String TRAP_L, String TRAP_angle, String CD_L,
String CD_angle, String distance_to_CD_X, String distance_to_CD_Y)
throws Exception {
    LoadLimits limits = new LoadLimits();
    passFlag=true;
    double ramp_l=Double.parseDouble(RAMP_L);
    if(ramp_l>limits.RAMP_L[limits.max_flag]||ramp_l<limits.RAMP_L[
limits.min_flag] ){
        Diagnostic diagnostic =new Diagnostic("Перевіряючий модуль");
        diagnostic.setStatus("Довжина рампи не відповідає встановленим
обмеженням__("+limits.RAMP_L[limits.min_flag]
+")__"+ramp_l+"__("+limits.RAMP_L[limits.max_flag] +")");
        passFlag=false;}

```

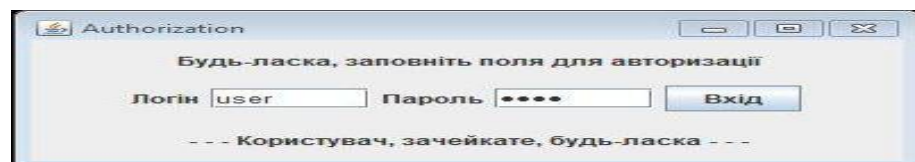
Коригуючий модуль розраховує значення довжини стулки залежно від введеного значення параметру довжини рампи.

Програмний код системи програмних модулів представлений у додатку В.

Результати авторизованого входу до розробленої системи представлені на рис. 4.14, де: а – Вхід з правами адміністратора до розробленої системи; б – Вхід з правами користувача до розробленої системи.



а – Вхід з правами адміністратора до розробленої системи



б – Вхід з правами користувача до розробленої системи

Рисунок 4.14 – Авторизація входу до розробленої системи

Варіант ведення БЗ в пакеті Protege 4.3, де показано обмеження, які накладаються на параметри в БЗ, зокрема, на параметри вантажного люка ЛА, представлено у додатку Б. Онтологічна схема зберігається як файл з розширенням .owl та відкривається за допомогою середовища IntelliJ Idea [200], використовуючи модуль JADE. Надалі створюються умови для відправлення або оновлення інформації в базі знань або прийняття деяких змін з БЗ. У додатку Б показано вікно прикладу java-класу Main для роботи з БЗ створення моделей СТО

в середовищі IntelliJ Idea.

При вході до системи з правами користувача є можливість тільки використання параметрів, закладених адміністратором в БД (рис. 4.13) та відпрацювання одного з обраних положень складових ВЛ ЛА.

№	Назва положення	Довжина рампи	Кут рампи	Довжина трапу	Кут трапу	Довжина ступки	Кут ступки	Початок ступки X	Початок ступки Y
1	cd_flight	3420	16.87	2040	90	6850	-19.2	9760	3210
2	cd_landing	3420	0	2040	0	6850	0	9760	3210
3	cd_land_squate	3420	-9.77	2040	7.4	6850	0	9760	3210
4	cd_land	3420	-14	2040	-2.3	6850	0	9760	3210
5	cd_land	3500	-14	2040	-2.3	6714	0	9760	3210

Рисунок 4.15 – Вхід до системи з правами користувача

При вході до системи з правами адміністратора є можливість тільки додавати положення ВЛ, змінювати значення параметрів в БД відповідно до обмежень з БЗ (рис. 4.14).

№	Назва положення	Довжина рампи	Кут рампи	Довжина трапу	Кут трапу	Довжина ступки	Кут ступки	Початок ступки X	Початок ступки Y
1	cd_flight	3420	16.87	2040	90	6850	-19.2	9760	3210
2	cd_landing	3420	0	2040	0	6850	0	9760	3210
3	cd_land_squate	3420	-9.77	2040	7.4	6850	0	9760	3210
4	cd_land	3420	-14	2040	-2.3	6850	0	9760	3210

Рисунок 4.16 – Вхід до системи з правами адміністратора

На рис. 4.17 показано інформаційне вікно, де викладені основні особливості проектування ВЛ, наведено рекомендації та досвід попередніх розробок ЛА. При внесенні змін значень параметрів ВЛ до системи (рис. 4.16) Перевіряючий модуль перевіряє відповідність введених даних значенням в БЗ та виводить повідомлення або дозволу подальшої роботи (рис. 4.18 а) або заборони (рис. 4.18 б). Корируючий модуль перераховує значення ступки ВЛ відповідно введеному значенню рампи ВЛ та видає про це відповідну інформацію (рис. 4.18 в).

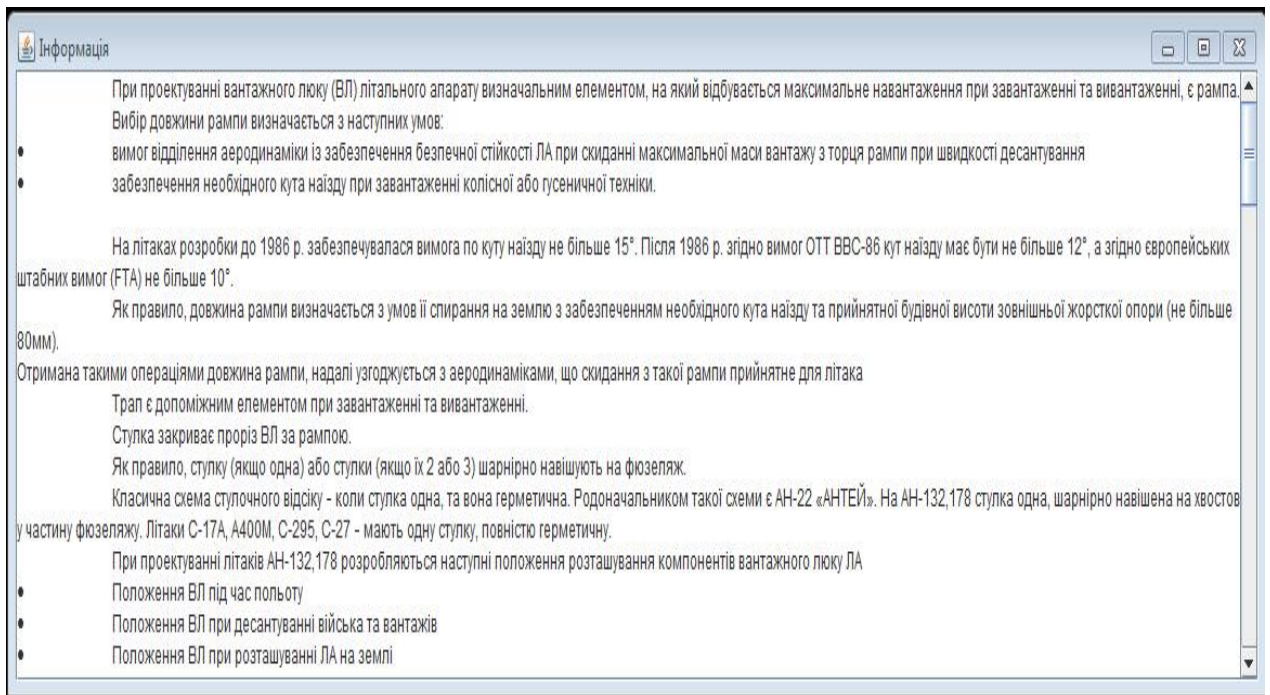
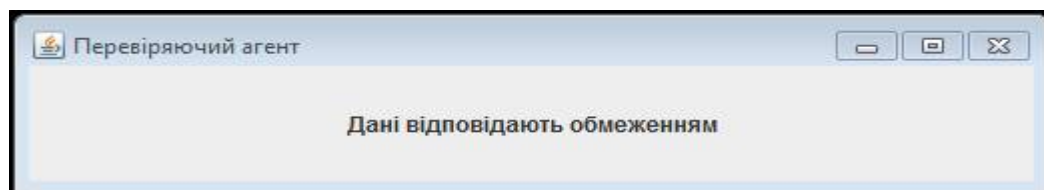
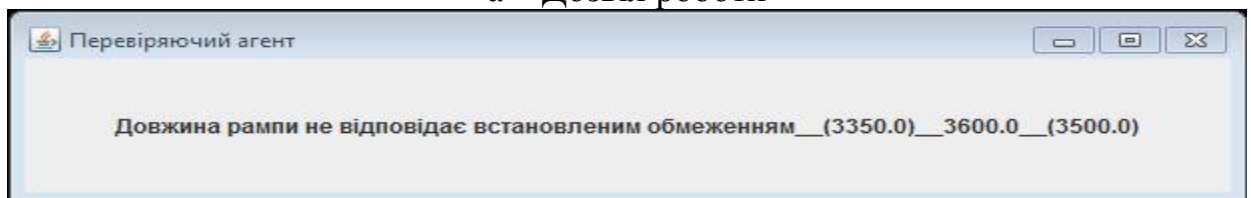


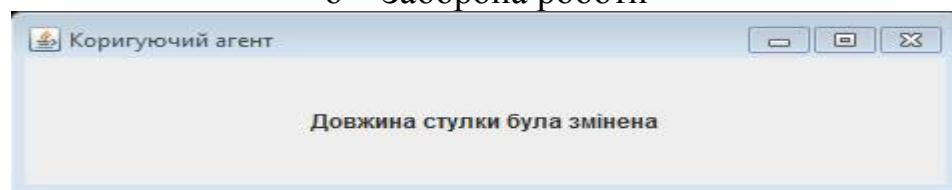
Рисунок 4.17 – Інформаційне вікно розробки вантажного люку



а – Дозвіл роботи



б – Заборона роботи



в – Інформація коригуючого модуля

Рисунок 4.18 – Результати роботи програмних модулів при внесенні змін значень параметрів

У роботі представлено програмний код реалізації нечіткого вислідку Мамдані з використанням мови Java для задачі побудови залежності параметрів вагової моделі: $m_{Wi} - p_{Wi}$ від параметрів геометричної моделі: $m_{Gi} - p_{Gi}$ та параметрів моделі міцності: $m_{Si} - p_{Si}$ компонента СТО (додаток В).

4.4 Приклад реалізації інформаційної технології з наявними складовими CALS та PLM-рішеннями та оцінка її ефективності

Вибір обчислювального середовища для розробки та підтримки реалізації інформаційної технології побудови моделей складних технічних об'єктів містить наступні напрямки підбору ПЗ: ІС САх та PLM; огляд мов високого рівня; вибір систем керування баз даних (СКБД); вибір онтологічної системи та середовища програмних модулів.

Внаслідок великих функціональних можливостей ІС TeamCenter та того, що ІС впроваджена (або впроваджується) на багатьох підприємствах аерокосмічної промисловості країн світу, що представляє інтерес як інструмент для реалізації інтегрованого процесу паралельного проектування виробів з підприємствами, на котрих вже впроваджено ІС TeamCenter. Зокрема, ІС PLM TeamCenter обрана фірмою Boeing в якості стандарту корпоративної PLM та використовується в практичній роботі провідними конструкторськими бюро та серійними заводами.

В роботі не ставилося за мету провести детальне порівняння функціональних можливостей ІС NX8 та CATIA V5, тому що ці ІС САх, безумовно, в наш час є визнаними світовими лідерами в області сучасних ІТ, але відомо також, що кожна з них має свої переваги та недоліки.

На рис. 4.19 наведено порівняння часу завантаження моделей в ІС САх NX та CATIA. Час завантаження при вирішенні типових задач на персональному комп'ютері (ПК) в середовищі NX8 та CATIA V5 показав приблизно однакові значення для простих (до 100 компонентів) та середніх (до 1000 компонентів) складальних одиниць, а при завантаженні складних складальних одиниць (від 1000 компонентів та більше) процес завантаження в середовищі NX8 відбувається швидше в 2-3 рази, ніж в CATIA V5. Розглянуто можливості використання мов програмування: C, C++, Visual Basic, Java, Python, PHP, Ruby, Perl, Паскаль, Delphi та інше. [201 – 204]

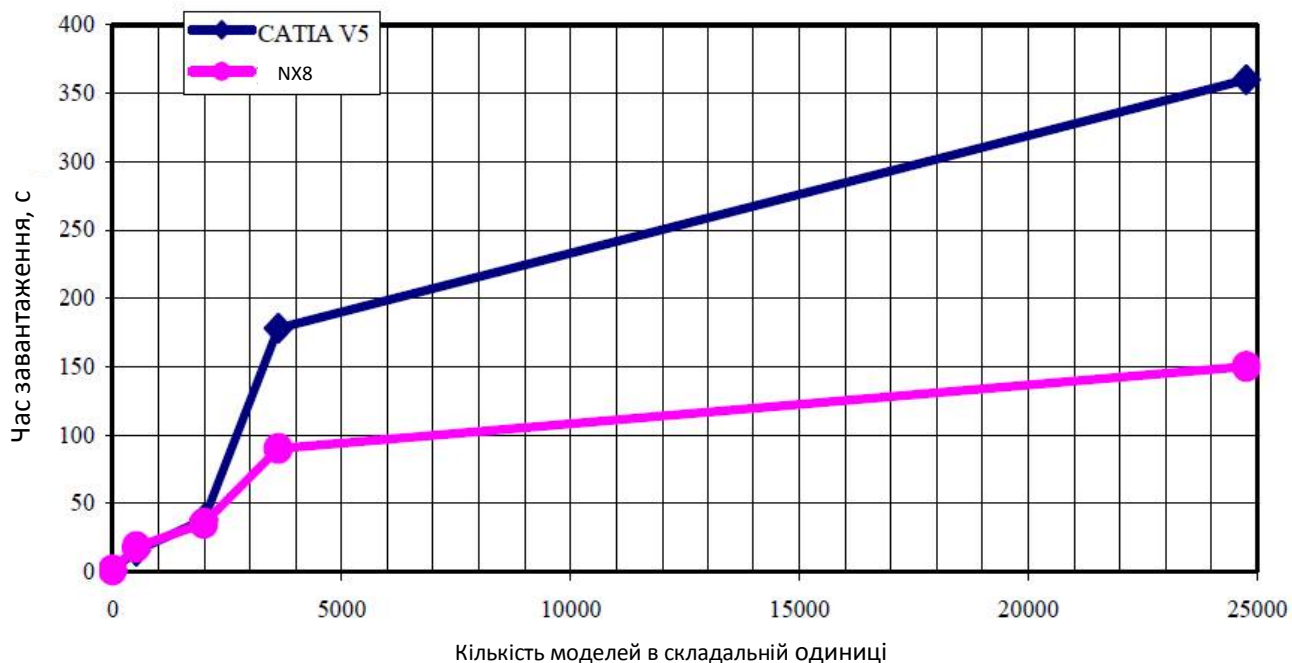


Рисунок 4.19 – Час завантаження моделі в ІС САх NX та САТІА

Оцінюючи функціональні можливості з розробки прикладного ПЗ в різних ІС САх, то NX8, на відміну від САТІА V5, має ширші можливості зі створення ПЗ для моделювання. Зокрема, це можливість використання NX Open – набору інструментів API (Application Programming Interface), який забезпечує гнучку інтеграцію додатків прикладного ПЗ з NX через відкриту архітектуру (тобто, об'єднання додатків з додатками NX для спільного використання даних на різних комп'ютерних платформах, в різномірних мережах, навіть через Інтернет). Крім того, окрім програмування на C/C++, Visual Basic.Net, C# та Java, NX представляє можливість використовувати технологію журнальних файлів шляхом протоколювання сесій NX (в вигляді скриптів Visual Basic.NET з подальшим редагуванням цих скриптів).

На основі вищезазначеного, при виборі ІС САх та PLM можна зробити наступний висновок: доцільно використовувати системи NX та САТІА V5 під загальним керуванням PLM TeamCenter.

Представимо схему реалізації БКС вантажного люку ЛА у вигляді кінематичної схеми в системі САх, де представлено чотири положення розташування основних параметрів ВЛ ЛА: лінії настилу рампи; лінії настилу ступки; лінії настилу трапу. Кінематична схема (КС) базової контрольної

структури вантажного люку ЛА посилається на БД "Параметри моделей СТО". Програмний код забезпечення зв'язку між САХ системою САТІА v5 та БД "Параметри моделей СТО" представлений у додатку Б.

Схема базової контрольної структури вантажного люку літального апарату в ІС САХ САТІА v 5 показана на рис. 4.20.

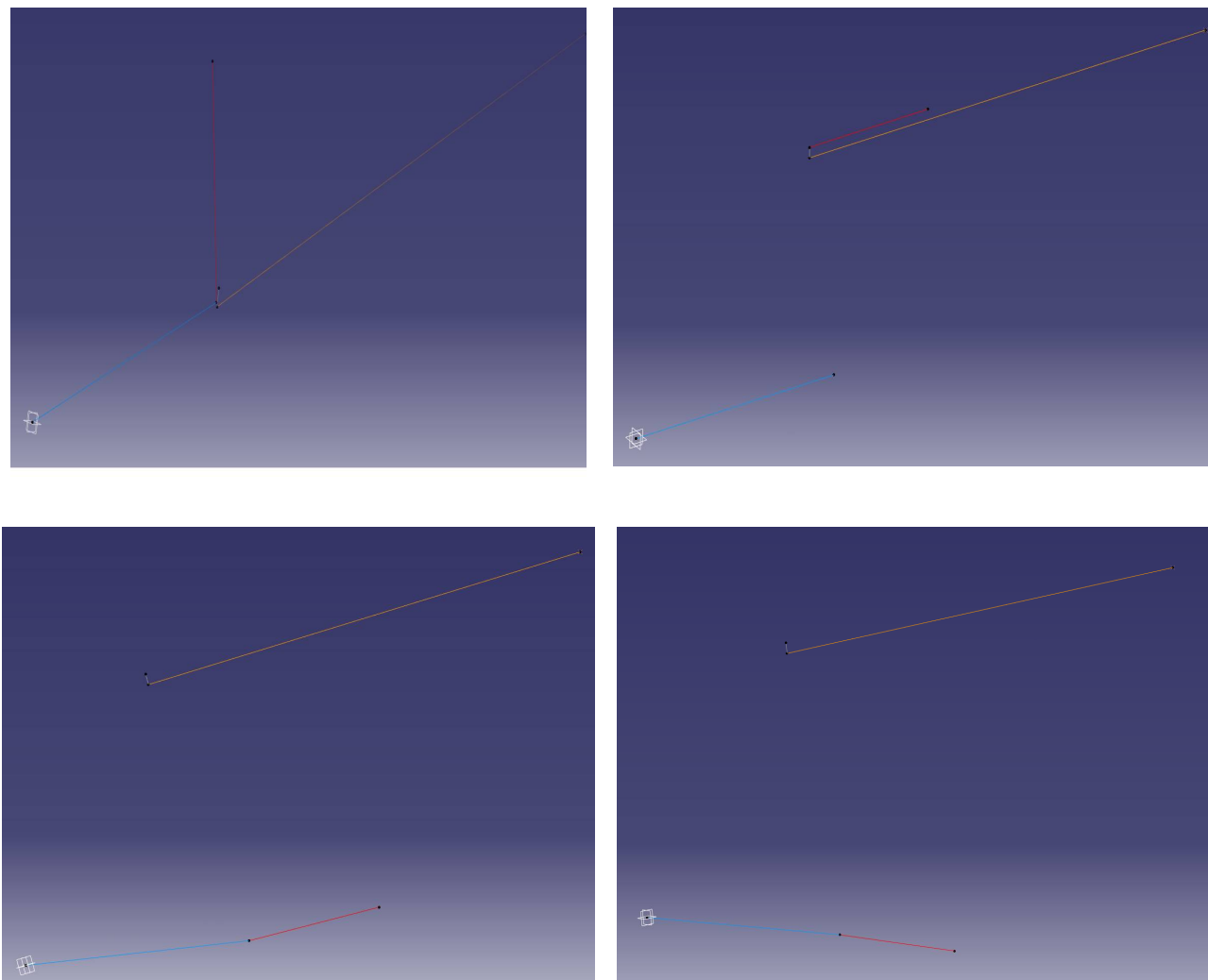


Рисунок 4.20 – Схема базової контрольної структури вантажного люку літального апарату в системі САХ САТІА v 5

Деталь базової контрольної структури (рис. 4.21) посилається на вищеперераховані параметри за рахунок опції "Копіювати зі зв'язком", включає в себе наступні складові: посилання на геометричні об'єкти ММГ або інші складові БКС, які використовуються при побудові даної моделі, допоміжну геометрію, необхідну для побудови БКС, базові точки прив'язки, базові лінії, направляючі лінії, площини, перетини площин або поверхонь, допустимі розміри довжин та

кутів, варіанти положень елементів агрегатів, побудованих на основі посилань на геометричні об'єкти ММГ або інших складових БКС, які використовуються при побудові даної моделі та на допоміжну геометрію, необхідну при побудові БКС.

На рис. 4.21 а показана БКС рампи ЛА [205] у вигляді кінематичної схеми вантажного люку ЛА, на рис. 4.21 б – схеми конструкції рампи ЛА, усі параметри якої зав'язані за побудовану кінематичну схему відповідно до описаної вище діаграми діяльності та на рис. 4.21 в розкрито обмін даними всередині базової контрольної структури. Зокрема, при зміні положення вантажного люку ЛА в таблиці автоматичні зміни надходять з кінематичної схеми в схему конструкції БКС вантажного люку ЛА.

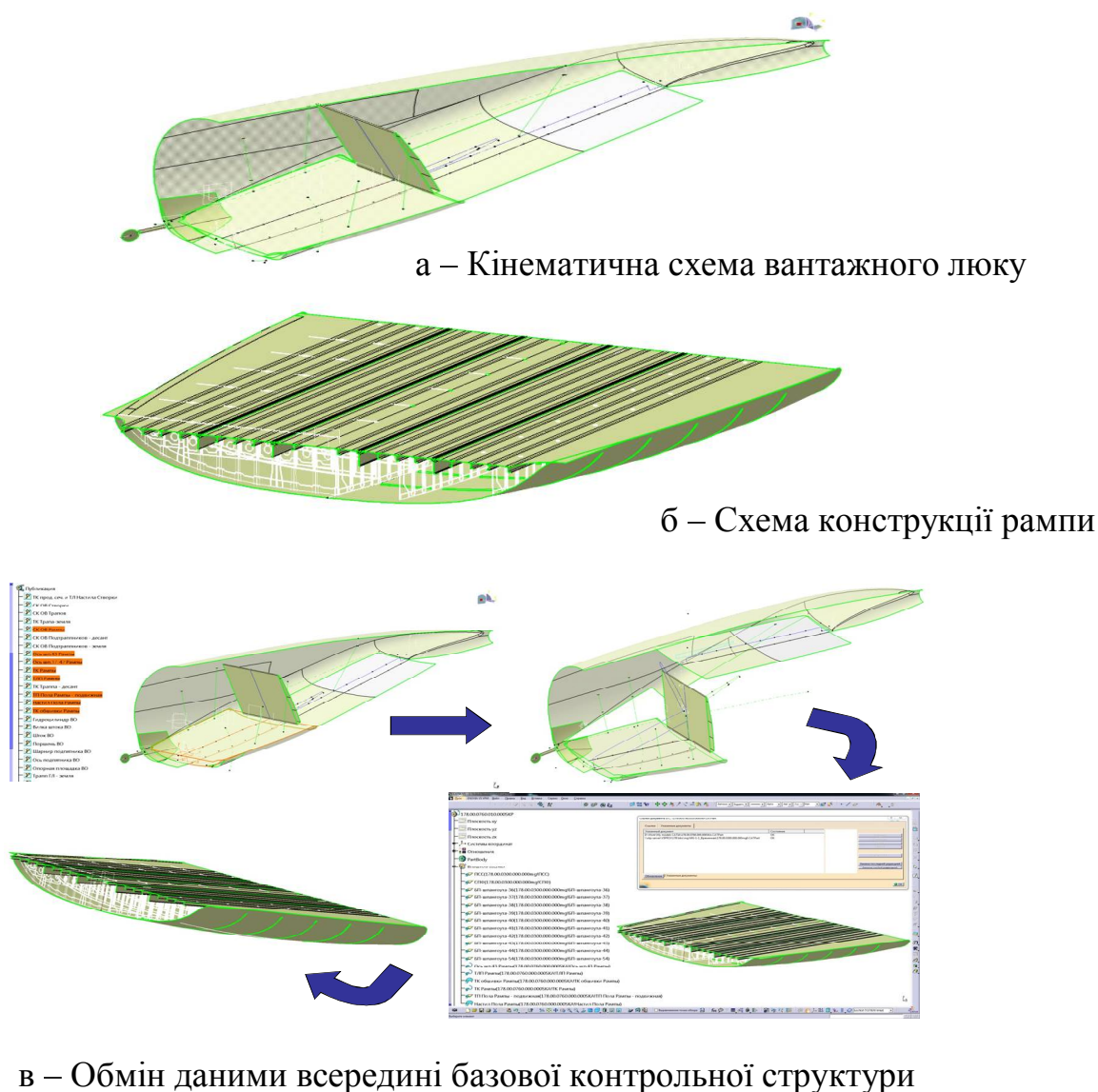


Рисунок 4.21 – Реалізація складових базової контрольної структури

На рис. 4.22 показана модель та дерево складальної одиниці моделі рампи ЛА на етапі МРО у вигляді твердотільної моделі, яка описує зовнішню форму і розміри, дозволяє оцінити її взаємодію з елементами оточення та служить для прийняття рішень при подальшій розробці виробу. В цій моделі представлено розташування основних складових даної частини ЛА: настил підлоги; обшивка рампи; поперечні ребра рампи (шпангоути). Ця модель дозволяє в першому наближенні провести розрахунки маси та координат центра мас рампи ЛА, також можливо використання даної моделі як на етапі МРО для узгодження з проектуванням систем та устаткування та проведення їх компонування, так і на етапі МПВ для подальшого уточнення даної моделі [44]. При зміні будь-яких параметрів частин в деталі БКС ЛА відбувається попередження в деталі МРО та при оновленні цієї деталі, зміни вступають в силу.

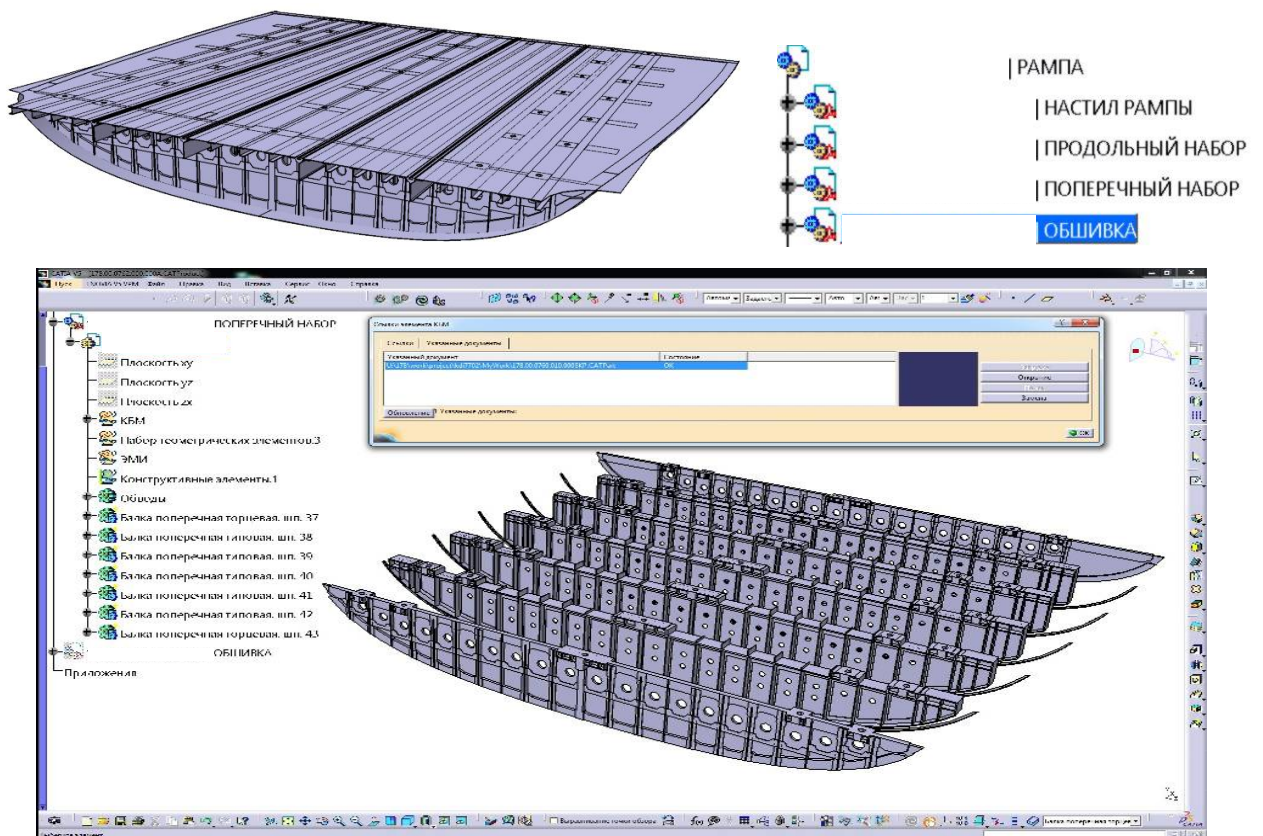


Рисунок 4.22 – Модель, дерево складальної одиниці моделі рампи літального апарату та склад поперечного набору рампи на етапі моделі розподілу об'єктів

На рис. 4.22 представлено склад моделей поперечного набору рампи ЛА на етапі розробки МРО та показано її зв'язок з схемою конструкції БКС рампи ЛА у

вигляді проробленої моделі, що відповідає етапу створення ескізного проекту та включає в себе балки поперечні по відповідному шпангоуту зі швартовочними вузлами та ободи рампи ЛА.

На рис. 4.23 представлено процес обміну даними всередині моделей відповідно до тих змін, які вносяться в БКС рампи ЛА. Після формування деталі вона буде займати своє відповідне місце в дереві проекту ЛА, створеному в САХ-середовищі на етапі розробки МРО та входить до складу відповідної складальної одиниці всередині даного дерева, яке відображає складові конструкції, систем та обладнання СТО, співвідношення між цими складовими частинами та умови їх входження до дерева проекту.

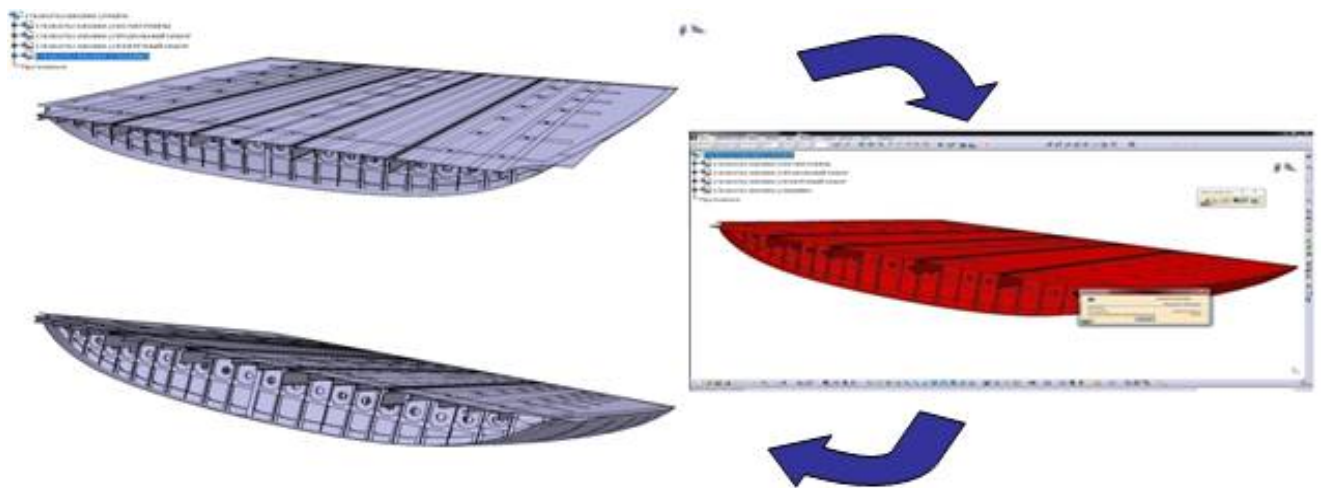


Рисунок 4.23 – Процес обміну даними всередині моделі відповідно до змін базової контрольної структури рампи літального апарату

Опишемо приклад створення моделей ВЛ ЛА з використанням ІС САХ – NX8. На рис. 4.24 представлено схему реалізації БКС вантажного люку ЛА у вигляді кінематичної схеми, де представлено чотири положення розташування основних параметрів ВЛ ЛА: лінія настилу рампи; лінія настилу трапу; лінія настилу ступки. Кінематична схема (КС) БКС вантажного люку ЛА посилається на БД. Програмний код зв'язку між NX 8 та БД представлено у додатку В.



Рисунок 4.24 – Схема базової контрольної структури вантажного люку в NX 8

На рис. 4.25 представлено БКС з САх системи NX8. На рис. 4.26 показано приклад моделей з САх системи NX8.

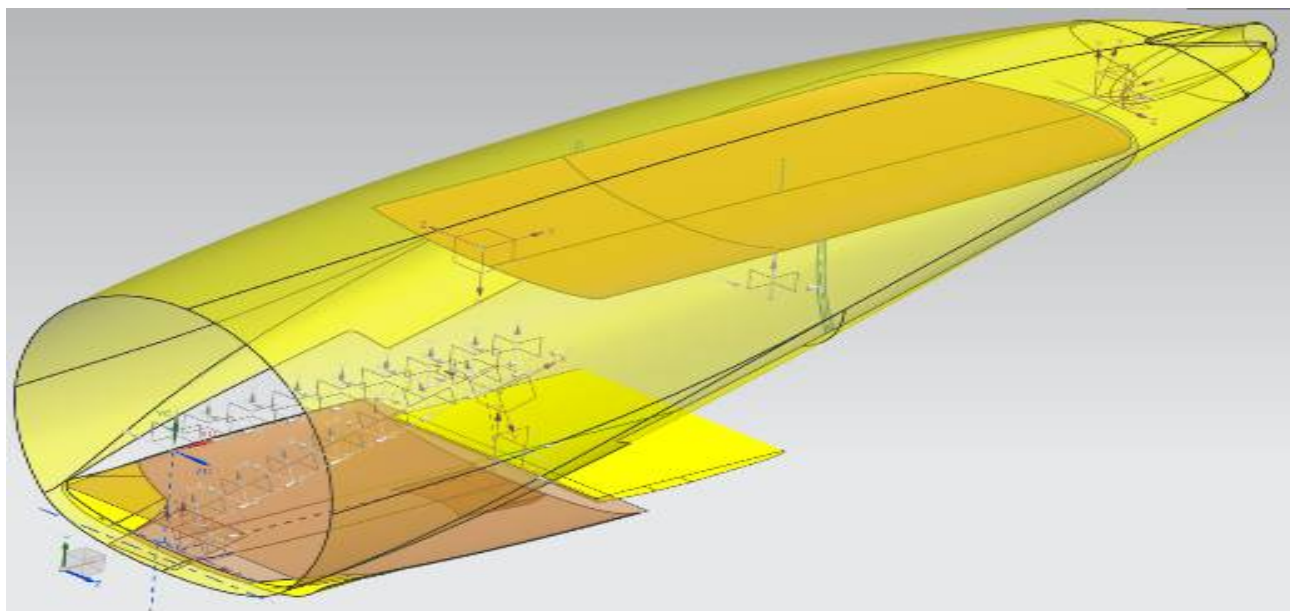


Рисунок 4.25 – Приклад базової контрольної структури з системи САх NX8

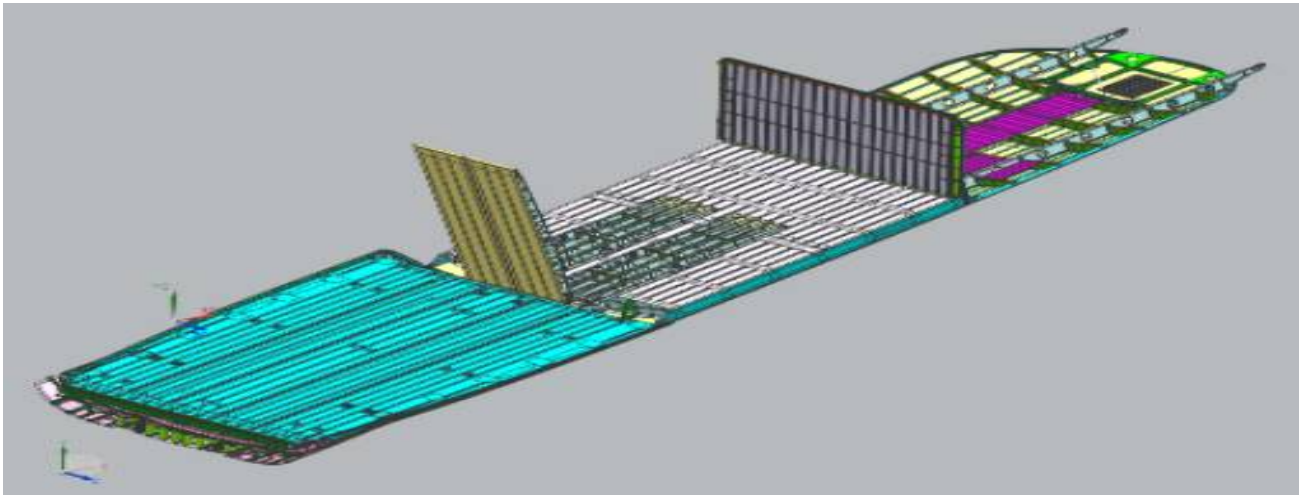


Рисунок 4.26 – Приклад моделі з системи CAx NX8

На рис. 4.27 представлено обмін даними між різноманітними моделями з різних ІС на різних стадіях створення СТО в ІС PLM TeamCenter.

№	Назва положення	Довжина рамп	Кут рамп	Довжина трапу	Кут трапу	Довжина ступки	Кут ступки	Початок ступки X	Початок ступки Y
1	cd_flight	3420	16.87	2040	90	8850	-19.2	9760	3210
2	cd_landing	3420	0	2040	0	8850	0	9760	3210
3	cd_land_squate	3420	-9.77	2040	7.4	8850	0	9760	3210
4	cd_land	3420	-14	2040	-2.3	8850	0	9760	3210

Рисунок 4.27 – Обмін даними між моделями в PLM TeamCenter

Завдяки ІТ можливо внести зміни в моделях виробу, що розроблюється, та відслідковувати відображення даних змін в структурі шляхом керування основними параметрами моделей в БД під контролем БЗ.

В табл. 4.3 представлено систему правил для ВЛ ЛА, в котрій розкрито відповідність дій програмних модулів в залежності від правил.

Таблиця 4.3 Система правил для ВЛ ЛА

Правило	Дія програмних модулів
ramp_l>limits.RAMP_L[limits.max_flag] ramp_l<limits.RAMP_L[limits.min_flag]	"Довжина рампи не відповідає встановленим обмеженням"
ramp_angle>limits.RAMP_angle[limits.max_flag] ramp_angle<limits.RAMP_angle[limits.min_flag]	"Кут рампи не відповідає встановленим обмеженням"
trap_L>limits.TRAP_L[limits.max_flag] trap_L<limits.TRAP_L[limits.min_flag]	"Довжина трапу не відповідає встановленим обмеженням"
trap_angle>limits.TRAP_angle[limits.max_flag] trap_angle<limits.TRAP_angle[limits.min_flag]	"Кут трапу не відповідає встановленим обмеженням"
cd_L>limits.CD_L[limits.max_flag] cd_L<limits.CD_L[limits.min_flag]	"Довжина ступки не відповідає встановленим обмеженням"
cd_angle>limits.CD_angle[limits.max_flag] cd_angle<limits.CD_angle[limits.min_flag]	"Кут ступки не відповідає встановленим обмеженням"
dist_to_CD_X>limits.distance_to_CD_X [limits.max_flag] dist_to_CD_X<limits.distance_to_CD_X [limits.min_flag]	"Відстань до точки обертання ступки (по X) не відповідає встановленим обмеженням"
dist_to_CD_Y>limits.distance_to_CD_Y [limits.max_flag] dist_to_CD_Y<limits.distance_to_CD_Y [limits.min_flag]	"Відстань до точки обертання ступки(по Y) не відповідає встановленим обмеженням"
CD_RAMP_check>limits.CD_CSYS_Limit_max CD_RAMP_check<limits.CD_CSYS_Limit_min	"Сумарна довжина трапу та ступки ("+limits.CD_CSYS_Limit_max+") не співпадає з вхідними даними ("+CD_RAMP_check+")"
newCD>limits.CD_L[limits.max_flag] newCD<limits.CD_L[limits.min_flag]	"Розрахована довжина ступки не відповідає встановленим обмеженням ("+limits.CD_L[limits.min_flag] +")"
else	"Довжина ступки була змінена"
passFlag	"Дані відповідають обмеженням"

Досліджено закономірності впливу зміни геометричних моделей компонентів ВЛ ЛА (рампи, ступки та трапа) на моделі компонування та центрування, вагову модель, та при скиданні вантажів.

Координата X_{T0} центру ваги завантаженого ЛА у % САХ розраховується за формулою: $X_{T0} = [(X_T - X_B) \cdot \cos \alpha_{CAH} + (Y_B - Y_T) \cdot \sin \alpha_{CAH}] \cdot \frac{100\%}{B_{CAH}}$,

де: X_B, Y_B – координати початку середньої аеродинамічної хорди (САХ) крила в системі координат ЛА; X_T, Y_T – координати центра ваги ЛА в системі координат ЛА; X_{T0}, Y_{T0} – координати центра ваги ЛА в системі координат САХ крила; B_{CAH} – довжина САХ; α_{CAH} – кут між проекцією САХ крила на площину симетрії ЛА та будівну горизонталь фюзеляжу (рис. 4.28).

Центр ваги першого вантажу знаходиться:

$$X_{T1} = \frac{M_{X_{T0}} + \sum_{i=1}^n M_{X_i}}{P_0 + \sum_{i=1}^n P_i} \cdot$$

Дані, котрі необхідні для розрахунку злітного центрування ЛА з вантажем представлено у табл. 4.4.

Таблиця 4.4 Дані для розрахунку злітного центрування ЛА

Найменування	PX_T	X_T	P	Y_T	PY_T
ЛА без вантажу	PX_{T0}	X_{T0}	P_0	Y_{T0}	PY_{T0}
Вантаж 1	PX_{T1}	X_{T1}	P_1	Y_{T1}	PY_{T1}
. . .					
Вантаж N	PX_{TN}	X_{TN}	P_N	Y_{TN}	PY_{TN}
ЛА з вантажем (до початку скидання)	$M_{X0} + \sum_{i=1}^n M_{X_i}$	X_{T1}	$P_0 + \sum_{i=1}^n P_i$	Y_{T1}	$M_{Y0} + \sum_{i=1}^n M_{Y_i}$

При скиданні, вантаж N зміщений на край рампи, його центр ваги в точці перевалювання (крайня точка рампи) знаходиться:

$$X_{T_2} = \frac{M_{X_{T_0}} + \sum_{i=1}^n M_{X_i} + P_N \cdot a}{P_0 + \sum_{i=1}^n P_i},$$

де a - відстань від X_{TN} до точки перевалювання; $P_N \cdot a = \Delta M_N$.

$$X_{T_2} - X_{T_1} = \frac{P_N \cdot a}{P_0 + \sum_{i=1}^n P_i}.$$

Виявлено вплив зміни розмірів рампи при скиданні одиночного вантажу:

$$\Delta \bar{X}_T = \left[\frac{P_N \cdot \Delta_P}{P_0 + \sum_{i=1}^n P_i} \right] \cdot \frac{\cos \alpha \cdot 100\%}{B_{CAH}}.$$



Рисунок 4. 28 – Схема САХ та система координат

Описано підхід до створення моделі технологічності в ІС PLM, використовуючи розроблену ІТ (рис. 4. 29).

При розробці методу знання-орієнтованої побудови моделей складного технічного об'єкта необхідно враховувати, що економічні, часові, обчислювальні та інші ресурси на весь ЖЦ СТО, завжди обмежені, а використані затрати для покращення кожної характеристики мають враховувати ці обмеження.

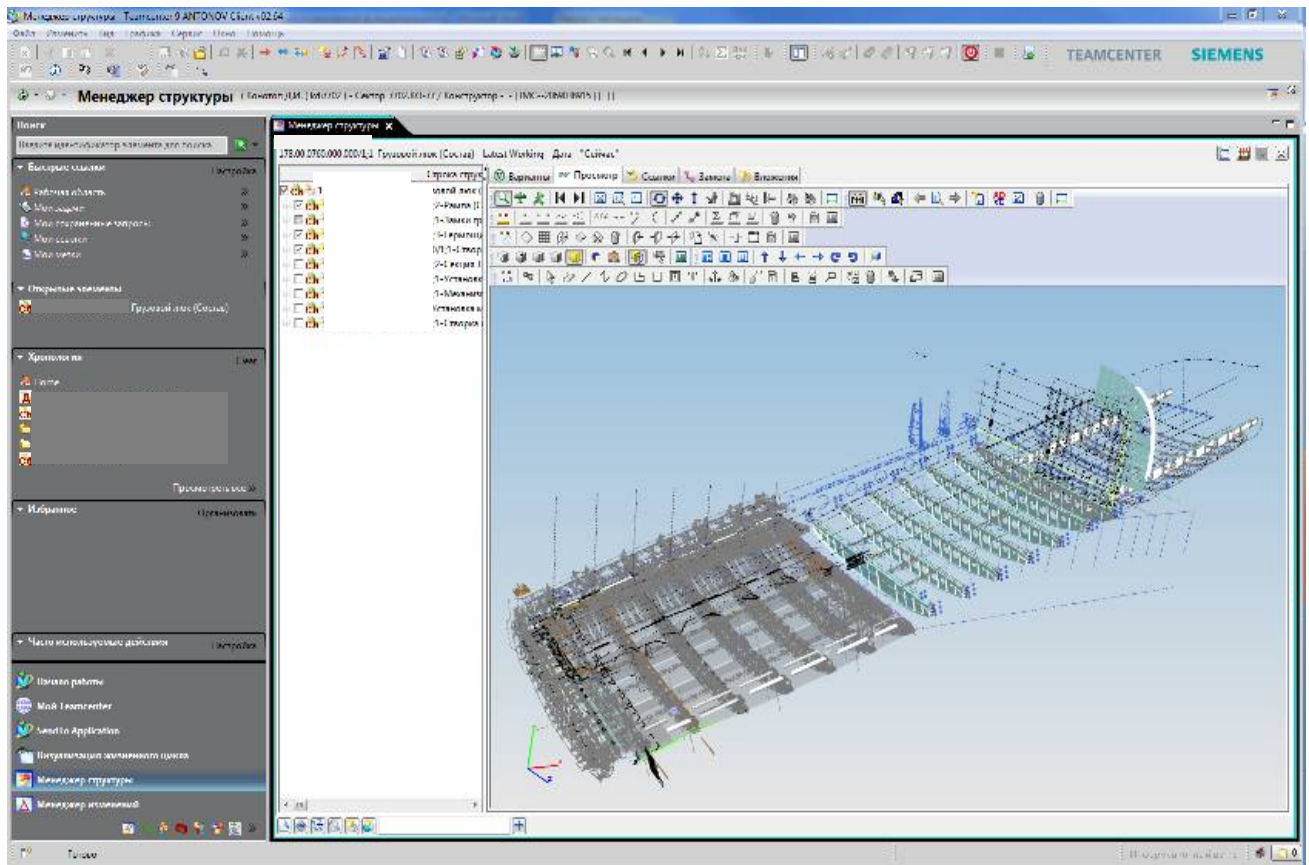


Рисунок 4. 29 – Приклад представлення моделі технологічності в IC PLM

Для раціонального розподілу цих ресурсів необхідно знати як відобразиться зміна затрат на покращення кожної характеристики якості створюваного продукту. Стандарт [205] найбільш повно регламентує технологію проектування, виробництва та забезпечення якості складних систем та продуктів.

Тривалість розробки моделей СТО є найважливішою економічною характеристикою створення СТО, оскільки часто вона визначає загальні строки розробки продукту, а тобто, швидкість реалізації проекту. На зменшення часу розробки моделей СТО значно впливає впровадження нових методів з організації процесу робіт зі створення СТО, які збільшують частку компонентів та операцій, які використовуються повторно. [206] При достатньо високому рівні технології велике значення має розмір та складність функціональних задач комплексів програм, а також якість, яка вимагається та відповідальність за створені програмні продукти. Багато часу також витрачається на узгодження конструктивних змін, запропонованих різними дисциплінами. [207]

Ефективність використання розробленої ІТ забезпечує скорочення строків проектування, зниження собівартості розробки СТО та підвищення його якості, оптимізує прийняття конструкторських рішень при узгодженні компонентів конструкції, систем і обладнання СТО вже на початкових етапах проектування за рахунок керування параметрами моделей, що створює швидкі та якісні умови для внесення змін в процес проектування та виробництва СТО.

Авторитетні джерела [4, 6, 208, 209] наводять середню тривалість проектного циклу створення ЛА 6 років. Важливо мати на увазі, що це не включає час розробки, що відводиться для ранньої конфігурації і аналіз ринку.

Згідно [210] вплив впровадження ЗОТ на основні етапи створення моделей СТО є дуже суттєвим. Порівняємо середній час створення моделей, використовуючи ІС САХ і PLM та описану ІТ(рис. 4.30). Час, який витрачається на розробку БД та БЗ з БКС звісно, перевищує час створення БКС окремо. Проте, відбувається суттєва економія часу створення моделей СТО за рахунок зменшення тривалості розробки етапів створення моделей СТО: МРО та МПВ.

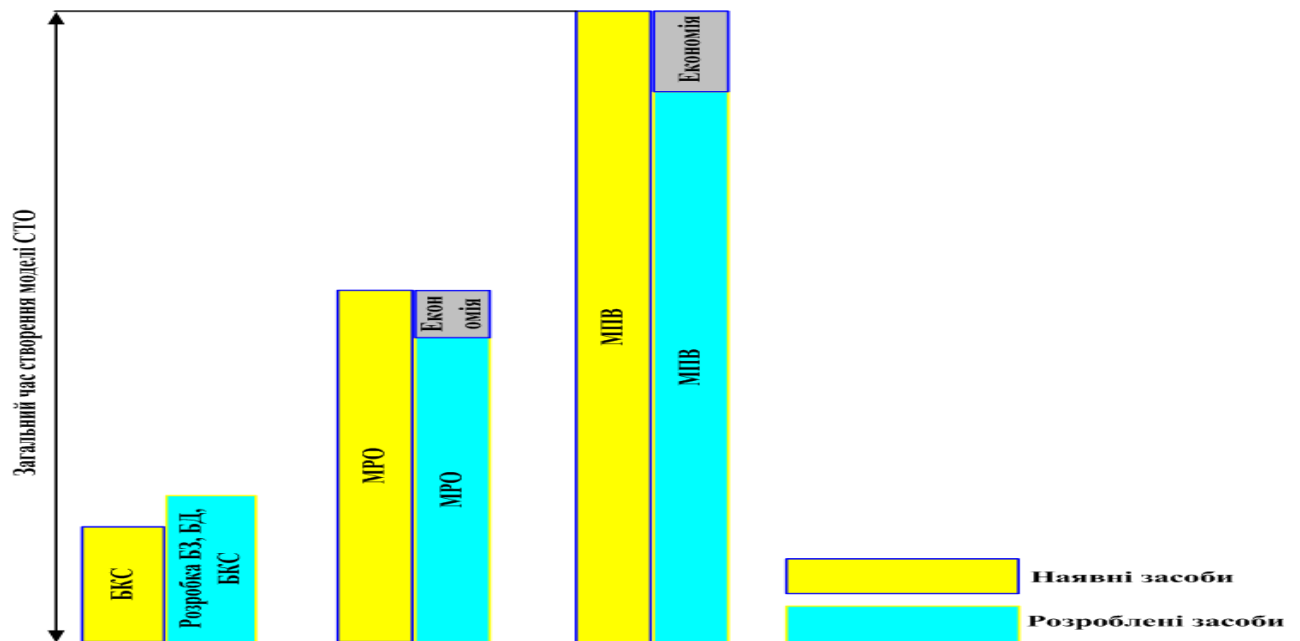


Рисунок 4.30 – Порівняння часу створення моделей складних технічних об'єктів наявними та розробленими засобами

Розглянемо приблизний середній час побудови моделей на прикладі вантажного люку ЛА. На основі визначення 1.1, модель ВЛ складається з: рампи,

трапу та стулки, що можна зобразити наступним чином: $M_{ВЛ} = \bigcup_{k=1}^3 M_k$. Моделі

ВЛ можна описати функцією взаємодії з БКС ВЛ: $M_{ВЛ} = f(P_{ВЛ})$. Модель рампи складається з: настилу рампи, поздовжнього та поперечного набору та обшивки,

що зобразимо: $M_P = \bigcup_{k=1}^4 m_k$.

У табл. 4.5 представлено приблизний середній час (в людино-годинах, лг) побудови моделей на прикладі вантажного люку ЛА. При розробці БД, БЗ та написанні програмного забезпечення виділяється програміст (адміністратор) з відділення попереднього проектування, який в подальшому буде також їх вести.

Таблиця 4.5 Середній час побудови моделей вантажного люку

Етап побудови моделей	Час побудови моделей існуючими методами, лг	Час побудови моделей розробленим методом, лг	
Розробка базової контрольної структури	160	160 + 100	Відділення попереднього проектування
Розробка моделі розподілу об'єктів	320	200	
Розробка моделі повного визначення	6800	5200	Відділення робочого проектування
Загальний час побудови	7280	5660	

Безумовно, важко точно оцінити ефект використання запропонованої ІТ. Та все ж, розрахунки показують близько 22 % економії матеріальних витрат. І це тільки розглядаючи модифікацію використання ІТ створення та супроводження узагальненої моделі СТО для керування параметричною інформацією. Якщо компанія використовує декілька ІС САх, тоді запропонована модифікація ІТ для обміну даними між різними складовими ІТ CALS та PLM-рішень, надасть додаткову ефективність використання ІТ.

4. 5 Висновки

1. Описано алгоритмізацію процесу створення інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів та обґрунтовано обрання програмних продуктів, які найбільше підходять для реалізації поставлених задач, використовуючи програмне забезпечення для реалізації бази знань, бази даних, модулів, складових інформаційних технологій CALS та PLM-рішень та зв'язку між ними.

2. Реалізовано інформаційну технологію керування параметричною інформацією шляхом використання бази знань для полегшення керування параметричних моделей у складному середовищі проектування складних технічних об'єктів, яка описує зв'язки, обмеження та правила між складовими процесу створення моделей складних технічних об'єктів та дозволяє зв'язати моделі з різних складових інформаційних технологій CALS та PLM-рішень.

3. Реалізовано зв'язок інформаційної технології керування параметрами моделей за допомогою програмних модулів. Розробка інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів здійснюється за допомогою середовища Java, в якому програмні модулі використовують інформацію про параметри моделей з бази знань.

4. Наведено приклад вирішення проектною задачі зв'язку моделей та параметрів баз даних та знань на прикладі створення моделей вантажного люку літального апарату.

5. Доведено ефективність використання розробленої інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів, що забезпечує скорочення строків проектування орієнтовно на 22%, зниження собівартості розробки складного технічного об'єкта та підвищення його якості, оптимізує прийняття конструкторських рішень при узгодженні моделей компонентів конструкції, систем і обладнання складного технічного об'єкта.

ОСНОВНІ РЕЗУЛЬТАТИ І ВИСНОВКИ

В дисертаційній роботі розв'язано актуальне науково-практичне завдання дослідження та розробки інформаційної технології створення та супроводження узагальненої моделі СТО з метою зниження термінів створення СТО та зменшення їхньої вартості при забезпеченні відповідної якості за рахунок налагодження зв'язку між моделями СТО та поєднання складових інформаційних технологій CALS та PLM-рішень. В роботі отримані наступні наукові результати:

1. Проведено системний аналіз сучасних технологій, методів і засобів створення моделей СТО та інформаційних технологій CALS та PLM-рішень, в результаті чого визначена необхідність розробки інформаційної технології створення та супроводження узагальненої моделі СТО.

2. Вперше сформульовано та досліджено узагальнену модель СТО, яка дозволяє налагодити інформаційний зв'язок між моделями СТО за рахунок використання для оцінки моделей СТО лінгвістичних змінних. В роботі розглянуто модель Мамдані для вирішення завдання побудови залежності параметрів геометричної моделі від параметрів моделі міцності та параметрів вагової моделі компонента СТО відповідно економічно обґрунтованим значенням обмежень вагової моделі та моделі міцності.

3. Вперше розроблено метод знання-орієнтованого створення СТО на базі узагальненої моделі, що дозволяє забезпечити взаємозв'язок між етапами створення складних технічних об'єктів, а також поєднати складові інформаційних технологій CALS та PLM-рішень за рахунок розробки наступних засобів: бази знань, що є основою для прийняття рішень в процесі створення СТО на основі знання-орієнтованих технологій; бази даних, що містить параметри моделей СТО; системи програмних модулів, яка підтримує зв'язок бази знань із базою даних.

4. Створено модифікації методу знання-орієнтованого створення СТО: керування параметричною інформацією при створенні моделей складних технічних об'єктів; обміну даними між складовими інформаційних технологій CALS та PLM-рішень для розв'язання завдань методу із відстеження зв'язків

даних на різних етапах побудови моделей СТО та поєднання компонентів технології CALS.

5. Розроблено інформаційну технологію створення та супроводження узагальненої моделі СТО та виконано оцінку її ефективності, що показало близько 22% економії матеріальних витрат при створенні агрегатів СТО на прикладі розробки моделей вантажного люку ЛА. У випадку проектної кооперації, коли використовується декілька інформаційних систем САх, буде отримана додаткова ефективність використання інформаційної технології за рахунок ведення одних баз знань та баз даних для всіх використаних інформаційних систем САх. Розроблена інформаційна технологія створення та супроводження узагальненої моделі та написане програмне забезпечення використовуються при проектуванні легкого транспортного літака та турбогвинтового регіонального вантажопасажирського літака на ДП «Антонов». Результати роботи впроваджені у навчальний процес КПІ ім. Ігоря Сікорського та Національного університету кораблебудування імені адмірала Макарова.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Antonenko O. Features of modeling failures of recoverable complex technical objects with a hierarchical constructive structure. / O. Antonenko S. Lenkov, G. Zhyrov and others. // Eastern-European Journal of Enterprise Technologies. – 2017. – № 4 (88). – P. 34–42.
2. Узунов О.В. Системне представлення складних технічних об'єктів в задачах аналізу та синтезу. / О.В. Узунов. // "Вісник НТУУ "КПІ". Серія машинобудування" – 2016. – №1 (76). – С. 126–132.
3. Raymer D. Aircraft design: A conceptual approach. Fourth Edition. / D. Raymer –American Institute of Aeronautics and Astronautics, 4th edition, 2006. – 869 p.
4. Основы технологии судостроения: Учебник / В. Л. Александров, Г. В. Бавыкин, А. С. Рашковский и др. Под общ. ред. В. Ф. Соколова. – СПб. : Судостроение, 1995. – 400 с.
5. Егер С. М. Проектирование самолетов: учебник для вузов / С. М. Егер, В. Ф. Мишин, Н. К. Лисейцев и др. Под ред. С. М. Егера. – М.: Машиностроение, 1983. – 616 с.
6. Hirz M. Integrated computer-aided design in automotive development. / M. Hirz, W. Dietrich, A. Gferrer, J. Lang // Development Processes, Geometric Fundamentals, Methods of CAD, Knowledge-Based Engineering Data Management. : Springer, Heidelberg – 2013. – P. 25–50.
7. Miller Frederic P. International Maritime Organization. / F. P. Miller, A. F. Vandome, J. McBrewster. – M Verlag Dr. Mueller e. K., 2010. – 318 p.
8. <https://www.icao.int>
9. Richter T. Integrated parametric aircraft design, Technical Report / T. Richter, M. Mechler, D. Schmitt // Institute of Aeronautical Engineering, Technical University Munich, 85747 Garching, 2002.
10. Frigg Roman. Models in Science / R. Frigg, S. Hartmann. [<http://Plato.Stanford.Edu/Archives/Fall2012/Entries/Models-Science/>]. In The Stanford

Encyclopedia of Philosophy. Edited by Edward N. Zalta. Stanford, CA: Stanford University Press, 2012.

11. Павлов А. А. Информационные технологии и алгоритмизация в управлении. / А. А. Павлов, С. Ф. Теленик – К.: Техніка, 2002. – 344 с.

12. ДСТУ 2938-94. Системи оброблення інформації. Основні поняття. Терміни та визначення

13. Боггс У. UML и Rational Rose. / У. Боггс, М. Боггс – М.: Лори, 2008. – 600 с.

14. Стоян Ю. Г. Математические модели и оптимизационные методы геометрического проектирования / Ю. Г. Стоян, С. В. Яковлев. – Киев: Наукова думка, 1986. – 259 с.

15. Timohin A. P. Providing tire stiffness tests in LMS Virtual.Lab-Imagine.Lab AMESim co-simulation. / A. P. Timohin, D. I. Konotop, V. P. Zinchenko // *Електроника и связь*. – 2012. № 1 – С. 42–48.

16. Моделирование параметров возмущенного движения самолета при сбросе груза в среде MATLAB / Цзинь Бин, В. П. Зинченко, С. В. Зинченко, Д. И. Конотоп [та ін.] // *Адаптивні системи автоматичного управління*. – 2012. – № 20(40). – С. 32–40.

17. Информационные технологии в наукоемком машиностроении: Компьютерное обеспечение индустриального бизнеса / Под общ. ред. А.Г. Братухина. – К.: Техніка, 2001. – 728 с.

18. Ковшов А. Н. Информационная поддержка жизненного цикла изделий машиностроения: принципы, системы и технологии CALS/ИПИ. / А. Н. Ковшов, Ю. Ф. Назаров, И. М. Ибрагимов, А. Д. Никифоров – М.: Академия (Academia), 2007. – 304 с.

19. Фісун М. Т. Аналіз та вибір моделей даних при створенні систем автоматизованого проектування / М. Т. Фісун, Є. О. Давиденко // *Збірник наукових праць НУК* – 2013. – №2(447). – С. 89–94.

20. Прогрессивные технологии моделирования, оптимизации и интеллектуальной автоматизации этапов жизненного цикла авиационных

двигателей: Монография / А. В. Богуслаев, Ал. А. Олейник, Ан. А. Олейник, Д. В. Павленко, С. А. Субботин; Под ред. Д. В. Павленко, С. А. Субботина. – Запорожье : ОАО "Мотор Сич", 2009. – 468 с.

21. Зінченко В. П. Інформаційні технології моделювання компоновки складного технічного об'єкта / В. П. Зінченко, Д. І. Конотоп, О. П. Сидоренко, В. В. Борисов // Інформаційні системи, механіка та керування. – 2011. – № 6. – С. 27–35.

22. Зінченко В. П. Застосування сучасних комп'ютерних інформаційних технологій при проектуванні складного технічного об'єкту / В. П. Зінченко, Д. І. Конотоп // Міжнародна науково-практична конференція "Інформаційні технології та комп'ютерна інженерія" (Вінниця, 19 - 21 травня 2010 р.) – Вінниця : ВНТУ – 2010. – Тези доповідей. – С. 50–51.

23. Процедура предварительного проектирования шасси транспортных самолетов / Линь Сунь, В. В. Борисов, В. П. Зинченко, Д. И. Конотоп // Інформаційні системи, механіка та керування – 2012 – № 8. – С. 60–70.

24. Конотоп Д.І. Визначення керуючої моделі проектування складного технічного об'єкта / Д.І. Конотоп // XII Міжнародна науково-практична конференція молодих учених і студентів «Політ. Сучасні проблеми науки» (Київ, 4 - 5 квітня 2012 р.) – К. : НАУ – 2012. – Тези доповідей. – С. 69.

25. Конотоп Д. И. Использование базовой управляющей структуры в процессе проектирования сложного технического объекта / Д. И. Конотоп // XIX Международная научная конференция студентов, аспирантов и молодых ученых «ЛОМОНОСОВ-2012» (Москва, МГУ имени М. В. Ломоносова, 9 - 13 апреля 2012 г.) – М. : Издательский отдел факультета ВМК МГУ – 2012. – Сборник тезисов. – С. 29–30.

26. Ванин И. В. Разработка компьютерных моделей базовой геометрии самолета с использованием современных информационных технологий. / И. В. Ванин, Г. А. Вирченко, Я. И. Ткачевский. // Открытые информационные и компьютерные интегрированные технологии – 2009. – № 42. – С. 82 – 86.

27. Ferguson David R. Mathematics for Industry: Challenges and Frontiers. A Process View: Practice and Theory / David R. Ferguson, Thomas J. Peters . – Society for Industrial and Applied Mathematics, SIAM, 2005. – 249 p.

28. Зинченко В. П. Средства и методы управления проектной информацией при создании сложных технических объектов / В. П. Зинченко, Ю. В. Абрамов, В. В. Борисов // Открытые информационные и компьютерные интегрированные технологии. – Харьков : Гос. Аэроком. Ун-т “ХАИ” – 2001. – № 9. – С. 17–29.

29. Абрамов Ю.В. Структура дерева сложного технического объекта как компонент технологии параллельного проектирования в среде CAD/CAM/CAE/PDM.// Открытые информационные и компьютерные интегрированные технологии. – Харьков : Нац. аэрокосм. ун-т им. Н. Е. Жуковского «ХАИ» – 2003. – № 17 – С. 17–30.

30. Абрамов Ю.В. Методологические основы формирования модели распределения пространства сложных технических объектов в среде CAD/CAM/CAE // Материалы международной конференции «Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта» – 2001. – М. : Станкин, 2001. – С. 70–82.

31. Метод интегрированного проектирования и компьютерного моделирования фюзеляжа гражданского самолета с помощью интегрированных систем CAD/CAM/CAE/PLM / А. Г. Гребеников, А. З. Двейрин, Ю. Н. Геремес, А. М. Гуменный // Открытые информационные и компьютерные интегрированные технологии : сб. науч. тр. / Харьков : Нац. аэрокосм. ун-т им. Н. Е. Жуковского «ХАИ» – 2006. – Вып. 30. – С. 10–30.

32. Sarh Branko. Aircraft manufacturing and assembly / B. Sarh (et al.) // Springer Handbook of Automation. Editor: Nof, Shimon Y. (Ed.) – 2009. – P. 893–910.

33. Andre Jean-Claude. From Additive Manufacturing to 3D/4D Printing. 1: From Concepts to Achievements / J.-C. Andre. – John Wiley & Sons, 2017 – 354 p.

34. Басов К. А. САТІА V5. Геометрическое моделирование. – М. : ДМК Пресс – СПб. : Питер, 2008. – 269 с.
35. Sarfraz Muhammad. Geometric Modeling: Techniques, Applications, Systems and Tools. / M. Sarfraz – Springer Science & Business Media, 2013 – 454 p.
36. Cox H. L. The Design of Structures of Least Weight / H. L. Cox // International Series of Monographs in Aeronautics and Astronautics: Solid and Structural Mechanics, Elsevier, 2014 – 144 p.
37. Абрамова Г. В. Алгоритми планування вагових розрахунків літака / Г. В. Абрамова, Д. І. Конотоп // Вісник Національного Авіаційного Університету. – 2013. – № 2 (55). – С. 129–135.
38. Abramova A. V. Application of Sheynin's plural calculations method at the initial stages of complex technical object design / A. V. Abramova, D. I. Konotop // 2-nd International Conference «Methods and Systems of Navigation and Motion Control» (Kyiv, October, 9 - 12, 2012) – К. : Освіта України – 2012. – Proceedings. – С. 129–132.
39. Египко В. М. Системы автоматизации экспериментальных исследований в аэродинамических трубах. / В. М. Египко, В. П. Зинченко, Б. Н. Белоусов, Ф. Н. Горин – К. : Наук. думка, 1992. – 264 с.
40. Barlow Jewel B. Low-Speed Wind Tunnel Testing, 3rd Edition. / J. B. Barlow, W. H. Rae, A. Pope –Wiley, 1999. – 713 p.
41. Underwood James R. Structural Design: A Practical Guide for Architects / J. R. Underwood, M. Chiuini – John Wiley & Sons, 2011. – 816 p.
42. Зінченко В. П. Проблеми оптимізації компоновки складного технічного об'єкту / В. П. Зінченко, С. В. Зінченко, Д. І. Конотоп, В. В. Борисов. // II наук. конференція магістрів та аспірантів, присвячена 20-річчю факультету прикладної математики (Київ, 14 - 16 квітня 2010 р.). Тези доповідей. – К. : НТУУ “КПІ” – 2010. – С. 64–67.
43. Голованов Н. Н. Геометрическое моделирование. – М. : Издательство Физико-математической литературы, 2002. – 472 с.

44. Павленко П. М. Автоматизовані системи технологічної підготовки розширених виробництв. Методи побудови та управління : монографія / П. М. Павленко. – К.: Книжкове вид-во НАУ, 2005. – 280 с.
45. Руднев В. И. Формирование технических объектов на основе системного анализа / В.И. Руднев [и др.] // М. : Машиностроение, 1991. – 320 с.
46. Тимченко А. А. Основи системного проектування та системного аналізу складних об'єктів. / А. А. Тимченко; [За ред. проф. В. І. Бикова]. 2-ге видання. – К. : Либідь, 2003. – 272 с.
47. Зеленский В. А. Проектирование сложных систем. / В. А. Зеленский. – Самара : Самар. гос. аэрокосм. ун-т им. С.П.Королева (нац. исслед. ун-т), 2012 – 96 с.
48. Зінченко В. П. Синтез структур і властивостей скінченноелементних моделей планера літака / В. П. Зінченко, В. В. Борисов // Наукові вісті Національного технічного університету України "Київський політехнічний інститут". – 2011. – № 1. – С. 62–68.
49. Тихомиров В. А. Разработка приложений для Unigraphics на языке C. В 2 ч. Учебное пособие / В. А. Тихомиров. – Комсомольск-на-Амуре : ФГБОУ ВПО "КНАГТУ", 2011. – 466 с.
50. Daintith John. A Dictionary of Computing (6 ed.) / J. Daintith, E. Wright – Oxford University Press, 2008 – 608 p.
51. Snow Charles C. Strategy, distinctive competence, and organizational performance. / Ch. C. Snow, L. G. Hrebiniak. // Administrative Science Quarterly. – Vol. 25, No. 2 – Jun. 1980, P. 317-336.
52. Brynjolfsson E. Beyond computation: Information technology, organizational transformation and business performance / E. Brynjolfsson, L. Hitt // Journal of Economic Perspectives – 2000. – 14 (4). – P. 23-48.
53. Journal of Information Technology Review – Volume: 9, Issue: 1 (February 2018). – 34 p.
54. <https://www.wiley.com/en-us>

55. Глушков В. М. Основы безбумажной информатики. / В. М. Глушков. – М.: Наука, 1982. – 552 с.
56. Дибкова Л. М. Информатика та комп'ютерна техніка: Посібник для студентів вищих навчальних закладів. / Л. М. Дибкова – К. : „Академвидав”, 2002. – 320 с.
57. Згуровский М. З. Системный анализ: проблемы, методология, приложения. / М. З. Згуровский, Н. Д. Панкратова. – К. : Наук. Думка, 2005. – 744 с.
58. Системний аналіз об'єктів та процесів комп'ютеризації: Навчальний посібник / А. В. Катренко. – Львів : Новий Світ-2000", 2003. – 424 с.
59. Маценко В. Г. Комп'ютерна графіка: Навчальний посібник. / В. Г. Маценко – Чернівці : Рута, 2009 – 343 с.
60. Пасічник О. Г. Основи веб-дизайну / О. Г. Пасічник, О. В. Пасічник, І. В. Стеценко: [Навч. посіб.]. – К. : Вид. група ВНУ, 2009. – 336 с.
61. Ситник В. Ф. Основи інформаційних систем : навч. посіб. / В. Ф. Ситник та ін. – Вид. 2-ге, перероб. і доп. – К. : КНЕУ, 2001. – 420 с.
62. Когаловский М. Р. Перспективные технологии информационных систем. / М. Р. Когаловский – М. : ДМК Пресс; Компания АйТи, 2003. – 288 с.
63. Mandel Michael I. Model-based expectation maximization source separation and localization. / M. I. Mandel, R. J. Weiss, D. P. W. Ellis. // IEEE Transactions on Audio, Speech, and Language Processing – February 2010 – 18(2) – 382–394 p.
64. ДСТУ 2392-94 Інформація та документація. Базові поняття.
65. ДСТУ 2941-94. Розроблення систем. Терміни та визначення.
66. <https://www.sita.aero/resources/blog/are-we-there-yet-are-we-there-yet/>
67. ГОСТ 27.001-95 Система стандартов "Надежность в технике".
68. Буч Гради. Объектно-ориентированный анализ и проектирование с примерами приложений (UML 2). Третье издание. / Гради Буч и др. – М. : «Вильямс», 2008. – 720 с.
69. Dawson F. ODA and Document Interchange / F. Dawson, F. Nielsen // UNIX Review. – 1990. – vol. 8, №. 3. – P. 50.

70. Норенков И. П. Информационная поддержка наукоемких изделий (CALS-технологии). / Норенков И. П., Кузьмик П. К. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2002. – 320 с.

71. Братухин А. Г. CALS (Continuous Acquisition and Life cycle Support – непрерывная информационная поддержка жизненного цикла изделия) в авиастроении / А. Г. Братухин, Ю. В. Давыдов, Ю. С. Елисеев, Ю. Б. Павлов, В. И. Суров. [Под ред. Братухина А. Г.] – М. : Изд-во МАИ, 2000. – 304 с.

72. ISO 9004-1. Управление качеством и элементы системы качества

73. Zeid Ibrahim. Mastering CAD-CAM / I. Zeid – Mcgraw-Hill Series in Mechanical Engineering, 2004. – 908 p.

74. Ли К. Основы САПР (CAD/CAM/CAE) / К. Ли – СПб. : Питер, 2004. – 560 с.

75. Rosenberg M. Elsevier's dictionary of technical abbreviations in English and Russian. / M.Rosenberg, S Bobryakov. – Amsterdam: Elsevier, 2005. – 1178 p.

76. Щеглов Д. К. Методы и средства выбора системы управления проектными данными конструкций летательных аппаратов / Научн. ред. О. А. Степанов; под общ. ред. В.Г. Пешехонова // Навигация и управление движением: Материалы докладов юбилейной X конференции молодых ученых «Навигация и управление движением». СПб. : ГНЦ РФ ЦНИИ «Электроприбор», 2009.

77. Ельцов М. Ю. Проектирование в NX под управлением Teamcenter: учеб. пособие / М. Ю. Ельцов, А. А. Козлов, А. В. Седойкин, Л. Ю. Широкова. – Белгород, 2013. – 752 с.

78. Fukuda Shuichi. Product lifecycle management for a global market / Sh. Fukuda, A. Bernard, B. Gurumoorthy, A. Bouras. // 11th IFIP WG 5.1 International Conference, PLM 2014, Yokohama, Japan – Springer, July 7-9 2014. – 520 p.

79. Зинченко В. П. Анализ средств и методов информационных технологий синтеза структур конечно-элементных моделей / В. П. Зинченко, В. В. Борисов, Д. И. Конотоп // Інформаційні системи, механіка та керування. – 2009. – № 3. – С. 112–121.

80. Зінченко В. П. Обчислювальні мережі: побудова та програмування: навч. посібник / В.П. Зінченко, С.В. Зінченко, І.В. Мірошніченко, Д.І. Конотоп, В. Резаї – К. : НТУУ «КПІ», 2011. – 240 с.
81. <http://www.cadcamcae.lv>
82. Brown Jim. Стратегии основных PLM-поставщиков в 2014 году и дальше. Часть I. Autodesk, Dassault Systèmes, PTC и Siemens PLM Software / J. Brown // CAD/CAM/CAE Observer – 2014 – № 1 (85) – С. 30–36.
83. www.autodesk.com
84. www.3ds.com
85. www.ptc.com
86. www.plm.automation.siemens.com
87. Мотовилов Д. Реализация типовых процессов разработки изделий в системе Pro/ENGINEER. / Д. Мотовилов // CAD/CAM/CAE Observer. – 2004. – № 4 (17).
88. <http://www.sterling.zp.ua>
89. Stokes M. Managing Engineering Knowledge МОКА: Methodology for Knowledge Based Engineering Application / M. Stokes – Professional Engineering Publishing, 2001. – 298 p.
90. Hepp M. Ontology management; semantic web / M. Hepp, P. de Leenheer, A. de Moor, Y. Sure // Semantic Web Services and Business Applications, Springer, 2008. – 295 p.
91. Kundu Ajoy Kumar. Aircraft Design. / A. K. Kundu – Queen's University Belfast, Cambridge University Press, 2010. – 650 p.
92. Абашев О. В. Применение искусственных нейронных сетей при проектировании самолетов. / О. В. Абашев, М. Ю. Куприков // Вестник МАИ. 2008. – Т. 15 №5 – С. 27–33.
93. www.fraunhofer.de – Fraunhofer Institute for Systems and Innovation Research, 2003.

94. Капитонова Ю. В. Математическая теория проектирования вычислительных систем / Ю. В. Капитонова, А. А. Летичевский. – М. : Наука, 1988. – 296 с.

95. Henderson R. P. Aircraft conceptual design for optimal environmental performance / Henderson R. P., Martins, J. R. R. A., and Perez, R. E. // *The Aeronautical Journal* – Jan. 2012 – Vol. 116, No. 1175 – P. 1–22.

96. Скурихин В. И. Теория и технология автоматизированного проектирования / В. И. Скурихин, В. В. Дубровский, В. Б. Шифрин, Н. Г. Бизюк – К. : Наук. думка, 1988. – 284 с.

97. Зинченко В. П. Методы и алгоритмы автоматизированного проектирования сложных технических объектов / В. П. Зинченко, В. В. Борисов // *Управляющие системы и машины*. – 2011. – № 1. – С. 46–56.

98. Борисов В. В. Исследование вопросов повышения качества разработки программного обеспечения PDM-систем за счет повышения независимости программных модулей / В. В. Борисов, Н. Ф. Горин, Н. П. Зинченко [и др.] // *Интеллектуальные информационно - аналитические системы и комплексы*. Собр. науч. тр. – К. : ИК им. В.М. Глушкова НАН Украины. – 2000. – С. 136–141.

99. Зинченко В. П. Проблемы оптимизации проектных исследований сложных технических систем / В. П. Зинченко // XXXV Междунар. симпозиум “Вопросы оптимизации вычислений”. – К. : ИК им. В. М. Глушкова НАН Украины. – 2009. Том. 1. – С. 253–259.

100. Конотоп Д. И. Оптимальное проектирование сложных технических объектов с использованием онтологического подхода / Д. И. Конотоп, В. П. Зинченко // *Научный журнал "Онтология проектирования"*. – Самара : "Новая техника". – 2011. – № 1 (2). – С. 44–53.

101. Белов Ю. А. Математическое обеспечение сложного эксперимента. Т. I. Обработка измерений при исследовании сложных систем / Ю. А. Белов, В. П. Диденко, Н. Н. Козлов [и др.] – К. : Наук. думка, 1982. – 304 с.

102. Краснощеков П. С. Последовательное агрегитирование в задачах внутреннего проектирования технических систем / П. С. Краснощеков,

В. В. Морозов, В. В. Федоров // Изв. АН СССР. Сер. технич. кибернет. – 1979. – №5. – С. 5–12.

103. Зінченко В. П. Апаратно-програмні комплекси для дослідження характеристик літаків / В. П. Зінченко, А. М. Гуржий // Наукові вісті НТУУ “КПІ”. – 2004. – № 3. – С. 43–55.

104. Борисов В. В. Проблемы обеспечения надежности функционирования PDM-систем / В. В. Борисов // Технології створення перспективних комп'ютерних засобів та систем з використанням новітньої елементної бази. Зб. наук. пр. – К. : ІК ім. В. М. Глушкова НАН України, 2000. – С. 67–72.

105. Управление общественными связями в бизнесе : учебник / А. Н. Загородников. – М. : КНОРУС, 2011. – 288 с.

106. Дружинин, В. В. Системотехника / В. В. Дружинин, Д. С. Конторов // М. : Радио и связь, 1985. – 200 с

107. Лямец В. И. Системный анализ. Вводный курс. / В. И. Лямец, А. Д. Тевящев // Харьков : ХНУРЭ, 1998. – 252 с.

108. Марка Д. А. Методология структурного анализа и проектирования / Д. А. Марка, К. МакГоуэн // М. : МетаТехнология, 1993. – 239 с.

109. Верников Г. Основы методологии IDEF1, IDEF1X, IDEF3, IDEF5 // <http://www.citforum.ru/cfin/vernikov>. Сервер информационных технологий “FORUM”

110. Perakath C. Benjamin et al. IDEF5 Method Report. Knowledge Based Systems, Inc, 1994. – 187 p.

111. Бондаренко М. Ф. Моделирование и проектирование бизнес-систем. Методы, стандарты, технологии / М. Ф. Бондаренко, С. И. Маторин, Е. А. Соловьева // Харьков : Компания СМІТ, 2004. – 272 с.

112. Йордан Э. Структурные модели в объектно-ориентированном анализе и проектировании. Пер. с англ. / Э. Йордан, К. Аргила // М. : Лори, 1999. – 264 с.

113. Михалевич В. С. Вычислительные методы исследования и проектирования сложных систем. / В. С. Михалевич, В. Л. Волкович // М. : Наука. Главная редакция физико-математической литературы, 1982. – 223 с.

114. Петров Э. Г. Методология структурного системного анализа и проектирования крупномасштабных ИУС. / Э. Г. Петров, С. И Чайников., А. О. Овезгельдыев // Харьков : Рубикон, 1997. – 140 с.

115. Вендров А. М. Проектирование программного обеспечения экономических информатизированных систем / А. М. Вендров // М. : Финансы и статистика, 2000. – 352 с.

116. Соловьева Е. А. Естественная классификация: системологические основания / Е. А. Соловьева // Харьков : ХТУРЭ, 1999. – 222 с.

117. Горбашко Е. А. Управление качеством: Учебное пособие. / Горбашко Е. А. // СПб. : Питер, 2008. – 384 с.

118. Mun D. Protection of intellectual property based on a skeleton model in product design collaboration / D. Mun et al. // Computer-Aided Design. – 2009. – vol. 41. – P. 641–648.

119. Smith R. Rapid airplane parametric input design / R. Smith et al. // Proc. of the 12th AIAA Computational Fluid Dynamics Conference, San Diego, California. – June 1995.

120. Athanasopoulos M. Parametric design of aircraft geometry using partial differential equations / M. Athanasopoulos et al. // Adv. Eng. Software. – 2009. – vol. 40. – P. 479–486.

121. Конотоп Д. І. Створення контрольної базової моделі в інформаційних технологіях проектування складного технічного об'єкта / Д. І. Конотоп, В. П. Зінченко // Наукові вісті Національного технічного університету України "Київський політехнічний інститут". – 2012. – № 6 (86). – С. 132–137.

122. Конотоп Д. І. Впровадження керуючої моделі в процес проектування складного технічного об'єкта / Д. І. Конотоп // «Наука і молодь» – К. : НАУ – 2012. – Тези доповідей. – С. 34. – 37.

123. <http://ts.nist.gov/standards/iges>

124. ISO TC184/SC4, ISO 10303-42-Part 42. Industrial automation systems and integration, 1994.

125. Burns Marshall. Automated Fabrication / M. Burns // Prentice Hall, 1993 – 369 p.

126. ISO 10303-203. Industrial automation systems and integration, 1994.

127. ISO 10303-111. Industrial automation systems and integration – Product data representation and exchange: Integrated application resource; 2007.

128. ISO International Standard 10303-11:1994, Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual, International Organization for Standardization, Geneva, Switzerland, 1994.

129. <https://www.khronos.org/webgl/> – WebGL - OpenGL ES 2.0 for the Web

130. Понятский В. М. Методика исследования ударных нагрузок конструкторских проектов САПР SolidWorks в среде имитационного моделирования MATLAB / В. М. Понятский, Г. И. Колесников, В. Г. Федорищева // «Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта» CAD/CAM/PDM-2008: Труды VIII междунар. конф. 21-23 октября 2008 г., М. : ИПУ, 2008. – С. 160–163.

131. Choi G. H. Exchange of CAD part models based on macro-parametric approach / G. H. Choi, D. Mun, S. Han // International Journal of CAD/CAM – 2002. – № 2(1). – P. 13–21.

132. Kim J. Standardized data exchange of CAD models with design intent / J. Kim, M. P. Pratt, R. G Iyer, R. D. Sriram // International Journal of CAD – 2008. – 40(7). – P. 760–777.

133. Yang J. An XML based macro data representation for a parametric CAD model exchange / J. Yang, S. Han, J. Cho, B. Kim, H. Y. Lee // Proceedings of the International CAD Conference and Exhibition, Pattaya Beach, Thailand – 2004. – P. – 153 – 162.

134. Seo T. S. Sharing CAD models based on feature ontology of commands history / T. S. Seo, Y. Lee, S. U. Cheon [and others] // International Journal of CAD/CAM – 2005 – № 5 – P. 39 – 47.

135. Bettig B. Derivation of a standard set of geometric constraints for parametric modeling and data exchange / B. Bettig, J. Shah // *International Journal of CAD* – 2001. – № 33. – P. 17–33.

136. Конотоп Д. І. Застосування онтології в обміні даними при розробці складного технічного об'єкта / Д. І. Конотоп // *Збірник доповідей ІХ Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки”* – Київ, 17 - 18 квітня 2013 – Секція 3. Інформаційні технології. – С. 437–440.

137. Палагин А. В. Системная интеграция средств компьютерной техники. Монография. / А. В. Палагин, Яковлев Ю. С. // *Винница : УНИВЕРСУМ-Винница, 2005. – 680 с.*

138. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В. Ф. Хорошевский – СПб. : Питер, 2001. – 384 с.

139. Gómez-Pérez Asunción. Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web / A. Gómez-Pérez, M. Fernández-López, O. Corcho // *Springer-Verlag London Limited, 2004. – 411 p.*

140. Gruber T. R. A Translation approach to portable ontology specifications. / T.R. Gruber // *Knowledge Acquisition* – 1993. – № 5. – P. 199–220.

141. Knublauch Holger. Weaving the Biomedical semantic web with the Protégé OWL plugin / H. Knublauch, O. Dameron, M. A. Musen // *Stanford Medical Informatics, Stanford University, Stanford, CA – 2004. – P. 5 – 6.*

142. Hotho A. Ontology-based text clustering / A. Hotho, A. Madche, S. Staab // *Proceedings of the IJCAI-2001. Workshop Text Learning: Beyond Supervision, Seattle, USA – 2001. – P. 48–54.*

143. Lee C. Automated ontology construction for unstructured text documents / C. Lee, Y. Kao, Y. Kuo, M. Wang // *Data & Knowledge Engineering* – 2007. – № 60. – P. 547–566.

144. T. Andreasen, P.A. Content-based text querying with ontological descriptors / T. Andreasen, P. A. Jensen, J. F. Nilsson [and others] // *Data & Knowledge Engineering*. – 2004. – № 48. – P. 199–219.

145. Gruninger M. *Ontology-applications and design* / M. Gruninger, J. Lee // *Communications of the ACM* – 2002. – Volume 45, Issue 2. – P. 39–41.

146. Guarino N. *Formal Ontology in Information Systems* / N. Guarino (ed.) // *Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998*. – Amsterdam : IOS Press – 1998. – P. 3–15.

147. https://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html – Noy Natalya F. *Ontology Development 101: A Guide to Creating Your First Ontology* / N. F. Noy, D. L. McGuinness // *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001*.

148. Конотоп Д. Застосування онтологій у процесі проектування літака / Д. Конотоп, Є. Абрамов, Г. Деркач // *Матеріали II Всеукраїнської конференції “Проблеми та перспективи розвитку української науки на початку третього тисячоліття”*. – Переяслав-Хмельницький, 2010. – С. 181–184.

149. Зінченко В. П. Аналіз можливості застосування методів штучного інтелекту в інформаційних технологіях проектування літаків / В. П. Зінченко, Д. І. Конотоп // *Збірник доповідей VIII Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки”* – Київ, 21-22 квітня 2011. – С. 66–71.

150. Зінченко В. П. Концепція застосування інтелектуальних технологій в проектуванні / В. П. Зінченко, Д. І. Конотоп, Г. В. Деркач, Є. Ю. Абрамов // *Открытые информационные и компьютерные интегрированные технологии* – Харьков : Гос. Аэроком. Ун-т “ХАИ” – 2011. – № 49. – С. 169–179.

151. Конотоп Д. І. Інтелектуальні технології проектування літальних апаратів. / Д. І. Конотоп // *Збірник тез доповідей науково – технічної конференції викладачів, науковців, аспірантів, студентів факультету авіаційних та космічних систем*. м. Київ – 2011. – С. 24.

152. Абрамов Є. Ю. Алгоритмізація бази знань проектування складного технічного об'єкту / Є. Ю. Абрамов, Д. І. Конотоп, Г. В. Деркач // XVIII Міжнародна конференція з автоматичного управління "Автоматика/Automatics – 2011". Львів, 28 – 30 вересня 2011 р. – Львів : Видавництво Львівської політехніки – 2011 – Матеріали конференції. – С. 358–359.

153. Abramova A. V. Using graphs for the planning at the initial stages of complex technical object design / A. V. Abramova, D. I. Konotop // The 2nd International Scientific Conference of Students and Young Scientists "Theoretical and Applied Aspects of Cybernetics" (Kyiv, November, 12 - 16, 2012) – К. : "Bukrek" – 2012. – Proceedings. – С. 8–13.

154. Abramova A. V. Ontology application for estimation of complex technical object characteristics / A. V. Abramova, D. I. Konotop // Електроніка та системи управління. – 2012. – №3 (33). – С. 81–88.

155. Abramov E. Knowledge-oriented support of complex technical object design / E. Abramov, D. Konotop, A. Abramova // 2-nd International Conference «Actual problems of UAV development» (Kyiv, October, 15 – 17, 2013) – 2013. – Proceedings. – P. 122–125.

156. Konotop D. Using ontology in geometrical modeling of complex technical object. / D. Konotop // VII International students and young scientists conference «Intelligence. Integration. Reliability» (Kyiv-Warsaw, April, 28 - 29, 2014) – К. : ІВЦ "Видавництво «Політехніка»" – 2014. – Abstracts. – P. 20–21.

157. Konotop D. Ontology using in geometrical models data processing of complex technical object / D. Konotop // XX-th International Conference «Knowledge-Dialogue-Solution» (Kyiv, September, 8 - 10, 2014) – ІТНЕА. Kyiv - Sofia – 2014 – Proceedings. – P. 118–119.

158. Конотоп Д. І. Використання знання-орієнтованих систем при створенні складного технічного об'єкта / Д.І. Конотоп, В.П. Зінченко, Лі Вей // Збірник доповідей X Міжнародної науково-технічної конференції "Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки" (Київ, 16 – 17 квітня 2015) – 2015. – Секція 4. Інформаційні технології. – С. 206–211.

159. <http://www.ksl.stanford.edu/software/ontolingua/>
160. Borgida A. Classic: A structural data model for objects / A. Borgida, R. Brachman, D. McGuiness, L. Resnick // ACM SIGMOID Int. Conf. on Management of Data, Portland, Oregon, USA – 1989. – P. 58 – 67.
161. Horridge M. A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools. Edition 1.0 / M. Horridge, H. Knublauch, A. Rector, R. Stevens, C. Wroe // University of Manchester, 2004. – 118 p.
162. <http://protege.stanford.edu>
163. Райордан Р. Основы реляционных баз данных / Райордан Р. // М. : Русская редакция, 2001. – 384 с.
164. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг // М. : Вильямс, 2003. – 1440 с.
165. Абрамов Ю. В. Опыт разработки и внедрения базы данных компонентов систем и оборудования сложных технических объектов в среде CAD/CAM/PDM. / Ю. В. Абрамов, О. П. Сидоренко // Сборник научных трудов Института Кибернетики им. В.М. Глушкова "НОВІ КОМП'ЮТЕРНІ ЗАСОБИ, ОБЧИСЛЮВАЛЬНІ МАШИНИ ТА МЕРЕЖІ", НАН України. – 2001. – Том 2. – С. 100–108.
166. Stark John. Product lifecycle management: 21st century paradigm for product realization / J. Stark // Springer-Verlag London Limited, 2011 – 560 p.
167. Уайлд Д. Оптимальное проектирование: Пер. с англ. / Д. Уайлд // М.: Мир, 1981. – 272 с.
168. Корн Г. Справочник по математике / Г. Корн, Т. Корн. – М.: Наука, 1978. – 832 с.
169. Кормен Т. Глава 15. Динамическое программирование / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн // Алгоритмы: построение и анализ. / Под ред. И. В. Красикова. – 2-е изд. – М. : Вильямс, 2005. – 1296 с.
170. Zadeh L. A. Fuzzy sets and possibility theory / L. Zadeh // New York: Pergamon Press, 1982. – 320 p.

171. Хорстманн Кей С. Java SE 8. Вводный курс / К. С. Хорстманн // М. : Вильямс, 2014. – 208 с.
172. <https://jena.apache.org/>
173. <http://semanticweb.org/wiki/Pellet>
174. Russell Stuart. Artificial Intelligence: A Modern Approach. 3rd edition / S. Russell, P. Norvig. // Pearson Publisher, 2009 – 1152 p.
175. Shoham Yoav. Multiagent systems: Algorithmic, Game-Theoretic, and Logical Foundations – Artificial Intelligence: A Modern Approach / Y. Shoham // Cambridge University Press, 2009. – 504 p.
176. Konotop D. Multi-agent-based conception of modern aircraft design / D. Konotop, I. Budinska, V. Zinchenko, E. Gatial // Proceedings of 5th Workshop on Intelligent and Knowledge Oriented Technologies (Bratislava, Slovakia, November 11 - 12, 2010) – 2010. – P. 125 – 128.
177. Endres-Niggemeyer Brigitte. Ontology-based information extraction in agents' hands. / B. Endres-Niggemeyer // KI-2008, OBIES-Workshop – 2008. – P. 18–26.
178. Nyulas Csongor. An Ontology-Driven Framework for Deploying JADE Agent Systems / C. Nyulas, M. J. O'Connor, S. W. Tu [and others] // IAT 2008 – 2008. – P. 573–577.
179. Okouya Daniel. Designing MAS organization through an integrated MDA/Ontology Approach / D. Okouya, L. Penserini, S. Sautrais // Proc. of the Transformation and Weaving Ontologies in Model Driven Engineering (MDE/MDA) – TWOMDE-08 @MoDELS'08, France – 2008. – P. 55 – 60.
180. Korhonen Jarmo. Using Web Services and Workflow Ontology in Multi-Agent Systems / J. Korhonen, L. Pajunen, J Puustjärvi // EKAW'02 workshop on Ontologies for Multi-Agent Systems – 2008.
181. Laclavik Michal. AGENTOWL: Semantic knowledge model and agent architecture / M. Laclavik, Z. Balogh, M. Babik, L. Hluchy // COMPUTING AND INFORMATICS – 2006. – Vol. 25, NO 5. – P. 421–439.

182. Li Yingguang. A semantics-based approach for collaborative aircraft tooling design / Y. Li, Y. Ruijie, J. Jianbang // *Advanced Engineering Informatics* – № 24(2). – 2010. – P. 149–158.

183. Li Yingguang. Aircraft tooling collaborative design based on multi-agent and PDM / Y. Li, Y. Ruijie, J. Jianbang, W. Liao // *Concurrent Engineering* – 2009. – vol. 17, №. 2 – P. 139–146.

184. Тамм Б. Г. Применение знаний в автоматизированных системах проектирования и управления / Б. Г. Тамм, Э. Х. Тыгу // *Прикладная информатика*. 1985. — Вып. 1 (8). – С. 5–25.

185. Добров Б. В. Онтологии и тезаурусы: модели, инструменты, приложения: монография / Б. В. Добров, В. В. Иванов, Н. В. Лукашевич, В. Д. Соловьев – Казань, 2008. – 220 с.

186. Палагин А. В. Реконфигурируемые вычислительные системы: Основы и приложения. / А. В. Палагин, В. Н. Опанасенко. – К. : Просвіта, 2006. – 280 с.

187. Палагин А. В. Онтологические методы и средства обработки предметных знаний: монография / А. В. Палагин, С. Л. Кривый, Н. Г. Петренко. – Луганск: изд-во ВНУ им. В. Даля, 2012. – 324 с.

188. Taniar D. *Web Semantics and Ontology* / D. Taniar, J. Rahayu / Idea Group Inc., London, 2006. – 404 p.

189. Garzas J. *Object-oriented design knowledge. Principles, Heuristics and Best Practices* / J. Garzas, M. Piattini // Idea Group Inc., London, 2007 – 362 p.

190. Йордан Э. Объектно-ориентированный анализ и проектирование систем / Э. Йордан, К. Аргила // Москва : ООО-Тиль, 2004. – 264 с.

191. Палагин А. В. Системная интеграция средств компьютерной техники. Монография / А. В Палагин, Ю. С. Яковлев // Винница : УНИВЕРСУМ-Винница, 2005. – 680 с.

192. Lukibanov O. Use of ontologies to support design activities in Diamler-Chrysler / O. Lukibanov // 8th International Protégé Conference – 2005.

193. <https://www.dbvis.com>

194. Rood Ronald. Mastering oracle scheduler in Oracle 11g Databases. / R. Rood // Packt Publishing; 1st edition, 2009. – 240 p.

195. Konotop D. I. 3D-models design concept of complex technical objects using knowledge-based technology/ D. I. Konotop, V. P. Zinchenko // Механіка гіроскопічних систем. – 2017. – № 34. – С. 5–13.

196. Конотоп Д.І. Знання-орієнтована побудова геометричних моделей складних технічних об'єктів / Д.І. Конотоп, В.П. Зінченко // Збірник доповідей XI Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки” (Київ, 13 - 14 квітня 2017) – 2017 – Секція "Інформаційні технології". – С. 72–75.

197. Конотоп Д. І. Створення та супроводження узагальненої моделі складних технічних об'єктів / Д. І. Конотоп // Збірник доповідей IX Міжнародної науково-технічної конференції “Інновації в суднобудуванні та океанотехніці” (Миколаїв, 18 - 19 жовтня 2018) – Секція "Інформаційні технології та управління проектами в промисловості, освіті та соціальній сфері". – С. 402–405.

198. Сеницкий Алексей. Грузовые авиаперевозки и их проблемы / А. Сеницкий // Логист – 2007. – №80.

199. <https://www.jadeworld.com>

200. <http://www.jetbrains.org>

201. Кормен Т. Алгоритмы. Построение и анализ. 2-е издание. / Т. Кормен, Ч. Лейзерсон, Р.Риверст, К.Штайн. М. : Вильямс, 2005. – 1296 с.

202. Ахо Альфред В. Структуры данных и алгоритмы. . Уч. пос. / А. Ахо, Д. Хопкрофт, Дж. Ульман. М. : Вильямс, 2007. – 400 с.

203. Седжвик Р. Фундаментальные алгоритмы на С++ / Р. Седжвик // К. : ДиаСофт, 2001. – 688 с.

204. Гудрич М. Т. Структуры данных и алгоритмы в Java. / М. Т. Гудрич // Мн. : Новое знание, 2003 – 671 с.

205. ISO 12207.2007

206. Друкер Питер Ф. Менеджмент.: Пер. с англ. / П. Ф Друкер, Дж. А. Макьярелло // М. : ООО “И.Д. Вильямс”, 2010. – 704 с.

207. Shekar Bharath. Managing complexity in aircraft design using design structure matrix / B. Shekar, R. Venkataram, B. M. Satish. // CONCURRENT ENGINEERING: Research and Applications – Volume 19. – 2011. – P. 283–294.

208. Spitz William. Development cycle time simulation for civil aircraft / W. Spitz, R. Golaszewski, F. Berardino, J. Johnson. – NASA/CR-2001-210658, 2001. – 92 p.

209. Mason William H. Modern Aircraft Design Techniques / W. H. Mason // Journal of Aircraft – 2003. – 37 p.

210. Skarka W. Application of MOKA methodology in generative model creation using CATIA / W. Skarka // Artificial intelligence – 2007 – 20 – P. 677–690.

ДОДАТКИ

А. Схеми створення складних технічних об'єктів

Б. Варіанти реалізації інформаційної технології

В. Лістинги програм

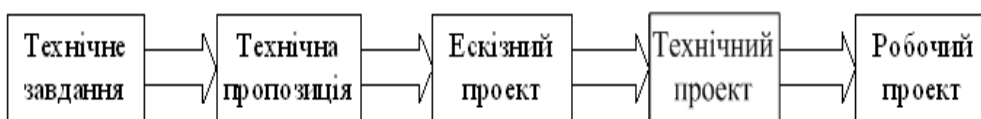
Г. Акти впровадження

Д. Список опублікованих праць за темою дисертації

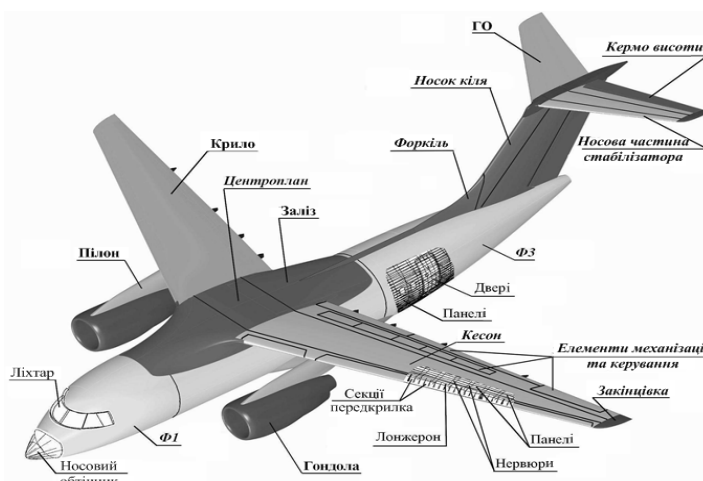
Додаток А

Схеми створення складних технічних об'єктів

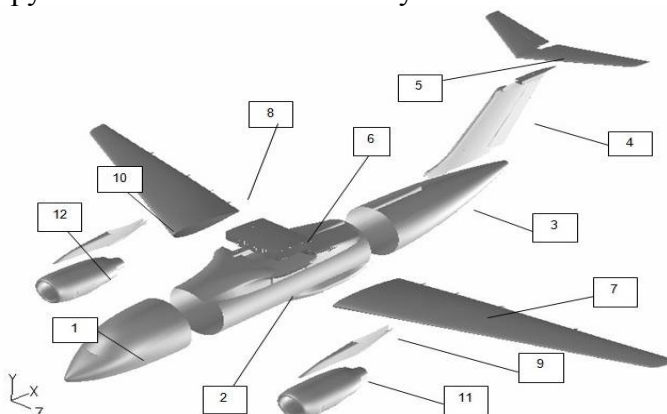
Стадії проектування складних технічних об'єктів



Приклад моделі майстер-геометрії складних технічних об'єктів на прикладі ЛА

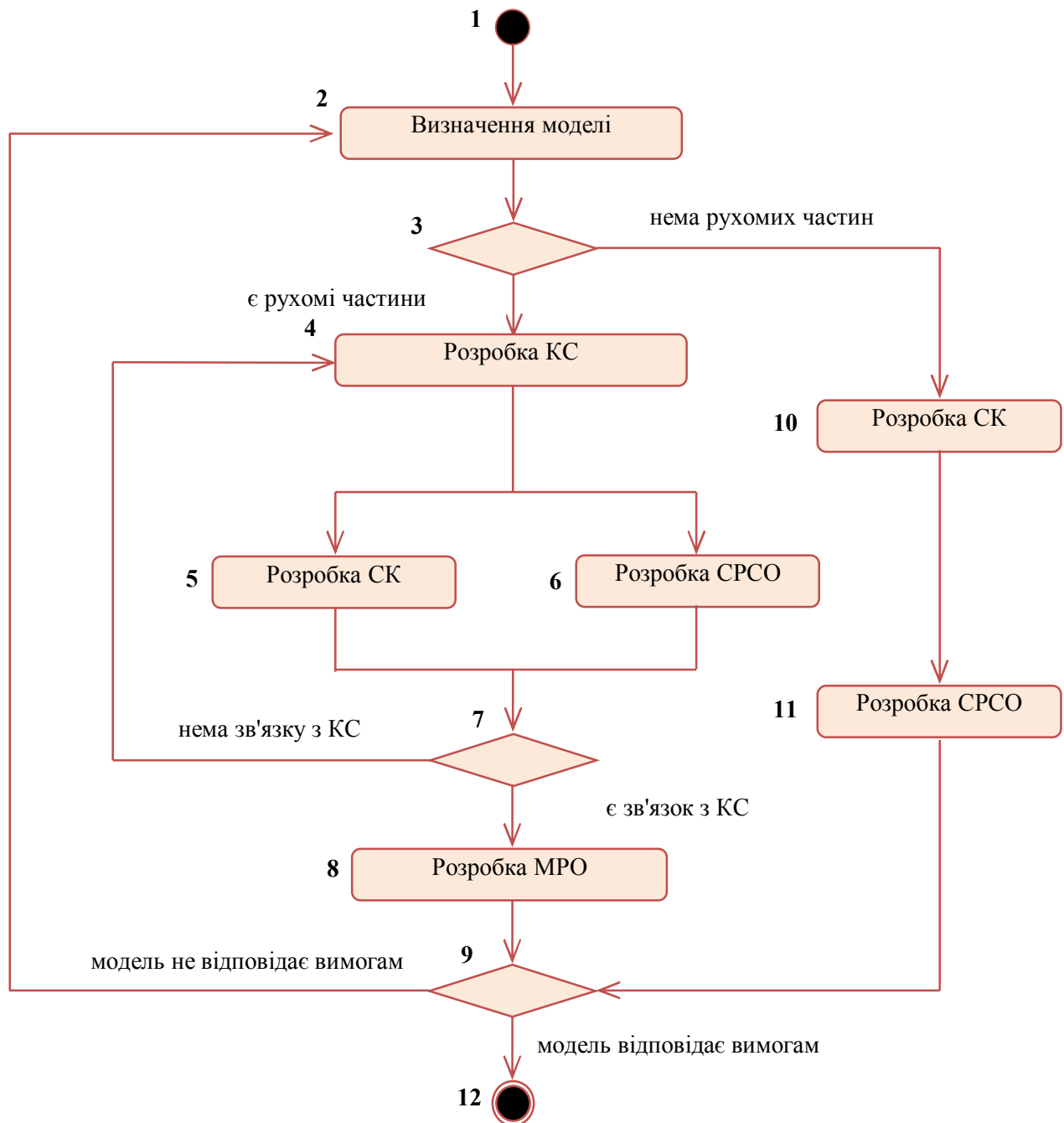


Конструктивно-технологічне членування літального апарату

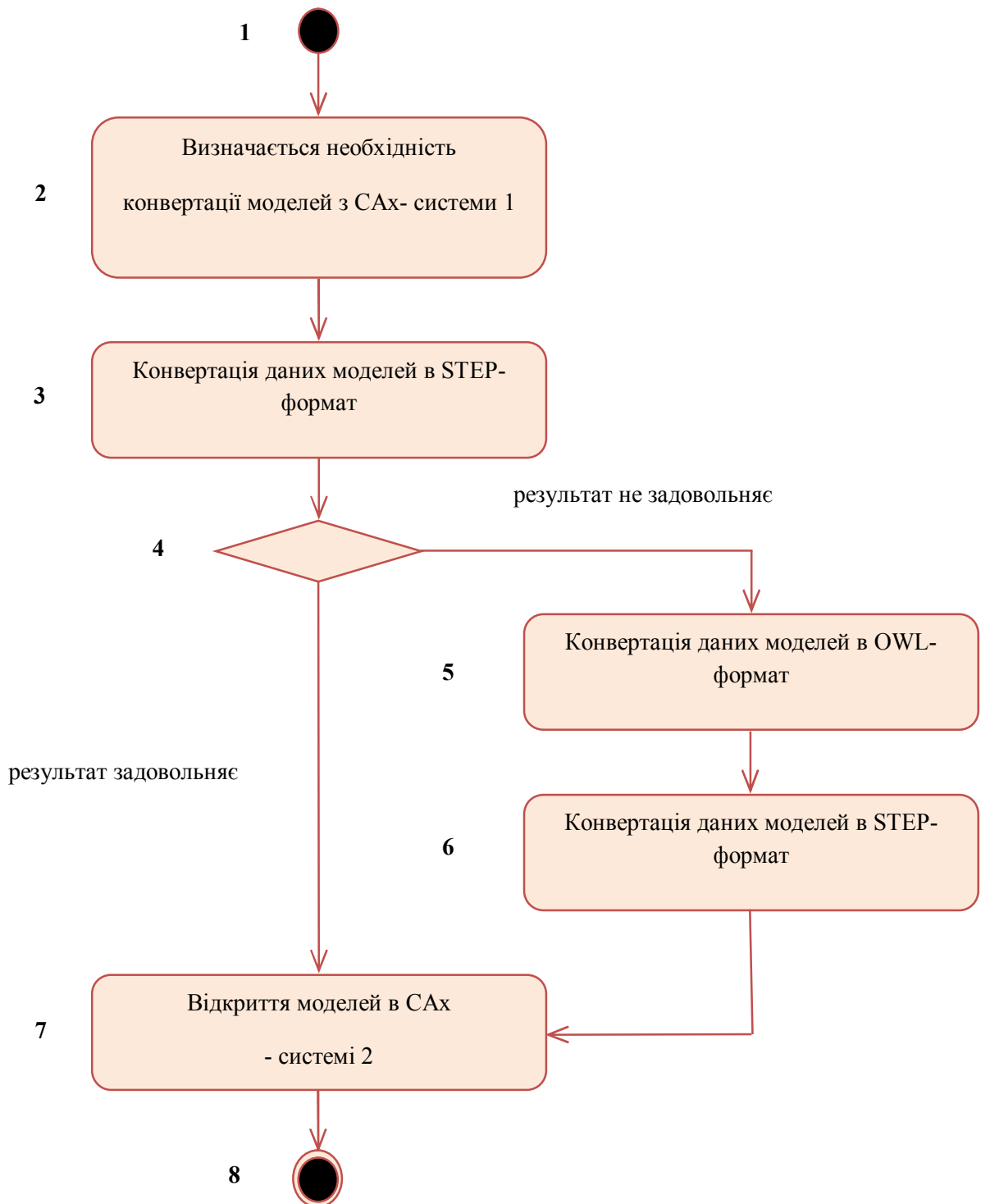


№	Найменування відсіку/ секції	Позначення
1	Носова частина фюзеляжу	Ф1
2	Мідельна частина фюзеляжу із залізом крила і обтічником шасі	Ф2
3	Хвостова частина фюзеляжу із залізами вертикального оперення	Ф3
4	Вертикальне оперення із кермами напрямку	ВО
5	Горизонтальне оперення з кермом висоти	ГО
6	Центроплан крила	ЦПК
7	Консоль крила ліва	ККл
8	Консоль крила права	ККп
9	Пілон лівий	Пл
10	Пілон правий	Пп
11	Силовa установка ліва	СУл
12	Силовa установка права	СУп

Діаграма діяльності процесу розробки моделей складних технічних об'єктів з керуванням параметрами



Варіант процесу обміну даними моделей з різних систем САх, використовуючи ЗОТ, представимо у вигляді діаграми діяльності



Список параметрів бази даних "Параметри моделей літальних апаратів"

<p>speed_liftoff – швидкість відриву;</p> <p>takeoff_run – довжина розбігу;</p> <p>approach_speed – швидкість заходу на посадку;</p> <p>path – довжина пробігу;</p> <p>airspeed_max – максимальна швидкість польоту;</p> <p>cruising_speed – крейсерська швидкість;</p> <p>flight_range_efficient – практична дальність польоту;</p> <p>rundown_range_efficient – перегінна дальність польоту;</p> <p>passenger_capacity – максимальна кількість пасажирів;</p> <p>flightrange_payload – дальність польоту (комерційне навантаження);</p> <p>cruising_altitude_max – максимальна висота крейсерського польоту;</p> <p>fuel_consumption – витрата палива;</p> <p>runway_length – довжина злітно-посадочної смуги;</p> <p>engine_count – кількість двигунів;</p> <p>engine_type – тип двигунів;</p> <p>bar_flight_max – максимальна злітна тяга;</p> <p>engine_behavior_bar – тяга на максимально-тривалій роботі двигунів</p>	ЛТХ
<p>making_cost – затрати на виготовлення ЛА;</p> <p>transportation_primecost – собівартість перевезення;</p> <p>operational_costs – експлуатаційні витрати;</p> <p>cost_A – вартість ЛА</p>	Економ- ічна модель ЛА
<p>payload_max – максимальне комерційне навантаження;</p> <p>take-off_mass_max – максимальна злітна маса;</p> <p>taxi_weight_max – максимальна маса руління;</p> <p>landing_weight_max – максимальна посадочна маса;</p> <p>fuel_weight_max – максимальна маса заправляваного палива</p>	Маса ЛА
<p>lg_b – продольна база шасі;</p>	Шасі ЛА

<p>lg_B – ширина колеї шасі; lg_H – висота шасі; lg_e – винос основних опор відносно центра мас (ЦМ) ЛА; lg_a – винос передньої опори відносно ЦМ ЛА; lg_f0 – кут перекидання; lg_f_park – кут стоянковий; lg_g – кут виносу подовжній основних опор відносно ЦМ ЛА; lg_c – кут виносу поперечний основних опор відносно ЦМ ЛА</p>	
<p>engine_d – діаметр двигуна; engine_l – довжина двигуна; gondola_d – діаметр гондоли двигуна; gondola_l – довжина гондоли двигуна</p>	СУ
tp_s – площа	ГО
vt_s – площа ВО	ВО
<p>wing_span – розмах крила; wing_s – площа крила; wing_aspect – подовження крила; wing_taper – звуження крила; wing_thickness_av – середня відносна товщина крила; incidence_angle – кут установлення; dihedral_angle – кут поперечного "V"</p>	Крило
<p>fuselage_d – діаметр фюзеляжу; fuselage_l – довжина фюзеляжу; mid_s – площа мідельного перерізу; f_extension – подовження фюзеляжу; f1_extension – подовження носової частини фюзеляжу; f3_extension – подовження хвостової частини фюзеляжу; fuselage_B – ширина фюзеляжу; fuselage_H – висота фюзеляжу</p>	Фюзеляж

<p>ramp_l – довжина рампи; trap_l – довжина трапу; cd_l – довжина стулки; ramp_angle – кут обертання рампи; trap_angle – кут обертання трапу; cd_angle – кут обертання стулки; distance_to_cd_x – відстань до осі обертання стулки по X;</p>	<p>Вантаж- ний люк (ВЛ)</p>
<p>distance_to_cd_y – відстань до осі обертання стулки по Y; cd_flight – положення ВЛ під час польоту; cd_landing – положення ВЛ при десантуванні війська та вантажів; cd_land – положення ВЛ при розташуванні ЛА на землі; cd_land_squate – положення ВЛ при розташуванні ЛА на землі "з присіданням"</p>	

Додаток Б

Варіанти реалізації інформаційної технології

1 Приклад обмежень параметрів вантажного люка літального апарату в базі знань в пакеті Protege 4.3

The screenshot shows the Protege 4.3 ontology editor interface. The main workspace displays a class hierarchy for an ontology. The classes are organized as follows:

- 'GM of C/D creation'
 - Attribute
 - 'Center of mass'
 - Mass
 - Name
 - Condition
 - 'Commoning scheme'
 - Models
 - Standards
 - 'Technical task'
 - Equipment
 - 'Trigon equipment'
 - 'Server equipment'
 - 'Geometrical model'
 - Airframe
 - 'Control parametric model'
 - Construction scheme
 - Kinematic scheme
 - 'Cargo_door_scheme'
 - cd_angle
 - cd_flight
 - cd
 - cd_land
 - cd_land_squate
 - cd_landing
 - distance_to_cd_x
 - distance_to_cd_y
 - ramp_angle
 - ramp_l
 - trap_angle
 - trap_l
 - Undercarriage_scheme
- Systems and equipment_scheme
 - power plant
 - 'Systems and equipment'

The right-hand pane shows the 'Description of' section for the selected class, 'Cargo_door_scheme'. It lists several constraints:

- Equivalent to: (none listed)
- Subclass of: (none listed)
- Subclass of (disjoint): (none listed)
- Creates some: (none listed)
- Uses some: 'Commoning scheme'
- Used some: Mass
- Used some: Standards
- Used some: 'Center of mass'
- Used some: 'Technical task'
- Members: (none listed)
- Target for key: (none listed)
- Disjoint with: (none listed)
- Disjoint union of: (none listed)

2 Приклад роботи з базою знань створення геометричних моделей складних технічних об'єктів, використовуючи агентів, в середовищі IntelliJ Idea

The screenshot displays the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the current file path: `Z:\1-Work\1-PHDTM4-Projects\H51 - Main.java - IntelliJ IDEA 2016.1.1`. The Project tool window on the left shows the project structure for `Z:\1-Work\1-PHDTM4-Projects`, including sub-projects like `H51` and `jade`. The main editor window displays the source code for `com.adt.H5Impl.class`.

```

package com.adt;

import com.adt.impl.*;
import com.adt.jin.Air_Condensation_System;
import com.adt.jin.Design_Recommendations;
import com.adt.jin.Technical_Task;
import sk.report.ontomap.OntoInterfaceImpl;

import java.util.LinkedList;
import java.util.List;

/**...*/
public class Main {
    public static void main(String[] args) throws NoSuchMethodException {
        OntoInterfaceImpl oint = new OntoInterfaceImpl("data/AD1_v3.owl");

        Air_Condensation_System acs = new Air_Condensation_SystemImpl("ACS D1");
        acs.setAme("ACS Unit 1");

        List container = new LinkedList();
        Technical_TaskImpl tt = new Technical_TaskImpl("TT1");
        tt.setAme("TT 1");
        tt.update();

        container.add(tt);
        Design_RecommendationsImpl dr = new Design_RecommendationsImpl("DR1");
        dr.setAme("REC1");
        dr.update();

        container.add(dr);
        acs.setUses(container);
        acs.update();

        for(Air_Condensation_System _acs : Air_Condensation_SystemImpl.listInstances(null, false)) {
            System.out.println("Name: "+_acs.getAme());
            System.out.print("Recommendations: ");
            for(Design_Recommendations dr : _acs.getUses(Design_RecommendationsImpl.class)) {
                System.out.print(" "+_dr.getAme()+" ", "");
            }
            System.out.println();

            System.out.println("List uses: ");
            for(Object _obj : _acs.getUses()) {
                System.out.print(" Name: ");
                if(!_obj instanceof Commoning_SchemeImpl) {
                    System.out.println(((Commoning_SchemeImpl)_obj).getAme());
                }
            }
        }
    }
}

```

The status bar at the bottom right indicates the file encoding is UTF-8 and the line number is 4245.

3 Лістинг програмної реалізації знання-орієнтованого середовища керування параметричною інформацією

```

<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.owl-ontologies.com/Ontology1281694117.owl#"
  xml:base="http://www.owl-ontologies.com/Ontology1281694117.owl"
  xmlns:sqwrl="http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:swrl="http://www.w3.org/2003/11/swrl#"
  xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
  xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:swrla="http://swrl.stanford.edu/ontologies/3.3/swrla.owl#">
<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://sqwrl.stanford.edu/ontologies/built-ins/3.4/sqwrl.owl"/>
  <owl:imports rdf:resource="http://swrl.stanford.edu/ontologies/3.3/swrla.owl"/>
  <owl:imports rdf:resource="http://protege.stanford.edu/plugins/owl/protege"/>
</owl:Ontology>
<owl:Class rdf:ID="Air_Condensation_System">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Air Condensation System</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Aircraft_Instrumentation">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>

```



```

<owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
<owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Aircraft Instrumentation</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Anti-Ice_System">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Anti-Ice System</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Auxiliary_Power_Plant">
  <rdfs:subClassOf rdf:resource="#Power_Plant"/>
  <owl:disjointWith rdf:resource="#Fire_Protection_System"/>
  <owl:disjointWith rdf:resource="#Fuel_System"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Control"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Installation"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Nacelle"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Auxiliary Power Plant</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Center_of_Mass">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#uses"/>
      <owl:someValuesFrom rdf:resource="#Geometrical_Model"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#ModelCharacteristics"/>
  <owl:disjointWith rdf:resource="#Mass_Characteristics"/>
  <owl:disjointWith rdf:resource="#Name_of_Model"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Center Of Mass</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Commoning_Scheme">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#used"/>
      <owl:someValuesFrom rdf:resource="#Geometrical_Model"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Documentation"/>
  <owl:disjointWith rdf:resource="#Data_for_Next_Stage"/>
  <owl:disjointWith rdf:resource="#Design_Recommendations"/>
  <owl:disjointWith rdf:resource="#Technical_Task"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Commoning Scheme</rdfs:label>
</owl:Class>

```

```

    >Commoning scheme</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Control_System">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Control System</rdfs:label>
</owl:Class>
<owl:ObjectProperty rdf:ID="created">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Data_for_Next_Stage"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <owl:inverseOf rdf:resource="#creates"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Equipment"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="creates">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Equipment"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <owl:inverseOf rdf:resource="#created"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Data_for_Next_Stage"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>

```

```

</owl:ObjectProperty>
<owl:Class rdf:ID="Data_for_Next_Stage">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#created"/>
      <owl:someValuesFrom rdf:resource="#Geometrical_Model"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Documentation"/>
  <owl:disjointWith rdf:resource="#Commoning_Scheme"/>
  <owl:disjointWith rdf:resource="#Design_Recommendations"/>
  <owl:disjointWith rdf:resource="#Technical_Task"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Aerodynamic data</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Design_Equipment">
  <rdfs:subClassOf rdf:resource="#Equipment"/>
  <owl:disjointWith rdf:resource="#Server_Equipment"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Design Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Design_Recommendations">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#used"/>
      <owl:someValuesFrom rdf:resource="#Geometrical_Model"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#used"/>
      <owl:someValuesFrom rdf:resource="#Name_of_Model"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Documentation"/>
  <owl:disjointWith rdf:resource="#Commoning_Scheme"/>
  <owl:disjointWith rdf:resource="#Data_for_Next_Stage"/>
  <owl:disjointWith rdf:resource="#Technical_Task"/>
</owl:Class>
<owl:Class rdf:ID="Documentation">
  <owl:disjointWith rdf:resource="#Equipment"/>
  <owl:disjointWith rdf:resource="#Geometrical_Model"/>
  <owl:disjointWith rdf:resource="#ModelCharacteristics"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Documentation</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Electric_System_Ensuring">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>

```

```

<owl:disjointWith rdf:resource="#Pressure_Control_System"/>
<owl:disjointWith rdf:resource="#Protection_Equipment"/>
<owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
<owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Electric System Ensuring</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Empennage">
  <rdfs:subClassOf rdf:resource="#Planer"/>
  <owl:disjointWith rdf:resource="#Fuselage"/>
  <owl:disjointWith rdf:resource="#Pylon"/>
  <owl:disjointWith rdf:resource="#Undercarriage"/>
  <owl:disjointWith rdf:resource="#Wing"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Empennage</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Equipment">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#creates"/>
      <owl:someValuesFrom rdf:resource="#Geometrical_Model"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <owl:disjointWith rdf:resource="#Documentation"/>
  <owl:disjointWith rdf:resource="#Geometrical_Model"/>
  <owl:disjointWith rdf:resource="#ModelCharacteristics"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Fire_Protection_System">
  <rdfs:subClassOf rdf:resource="#Power_Plant"/>
  <owl:disjointWith rdf:resource="#Auxiliary_Power_Plant"/>
  <owl:disjointWith rdf:resource="#Fuel_System"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Control"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Installation"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Nacelle"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Fire Protection System</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Freight_Equipment">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Freight Equipment</rdfs:label>
</owl:Class>

```

```

<owl:Class rdf:ID="Fuel_System">
  <rdfs:subClassOf rdf:resource="#Power_Plant"/>
  <owl:disjointWith rdf:resource="#Auxiliary_Power_Plant"/>
  <owl:disjointWith rdf:resource="#Fire_Protection_System"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Control"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Installation"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Nacelle"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Fuel System</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Furnishings">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Furnishings</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Fuselage">
  <rdfs:subClassOf rdf:resource="#Planner"/>
  <owl:disjointWith rdf:resource="#Empennage"/>
  <owl:disjointWith rdf:resource="#Pylon"/>
  <owl:disjointWith rdf:resource="#Undercarriage"/>
  <owl:disjointWith rdf:resource="#Wing"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Fuselage</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Geometrical_Model">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#used"/>
      <owl:someValuesFrom rdf:resource="#Center_of_Mass"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#uses"/>
      <owl:someValuesFrom rdf:resource="#Commoning_Scheme"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#creates"/>
      <owl:someValuesFrom rdf:resource="#Data_for_Next_Stage"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#uses"/>

```

```

    <owl:someValuesFrom rdf:resource="#Design_Recommendations"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#used"/>
    <owl:someValuesFrom rdf:resource="#Mass_Characteristics"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#uses"/>
    <owl:someValuesFrom rdf:resource="#Technical_Task"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#created"/>
    <owl:someValuesFrom rdf:resource="#Equipment"/>
  </owl:Restriction>
</rdfs:subClassOf>
<owl:disjointWith rdf:resource="#Documentation"/>
<owl:disjointWith rdf:resource="#Equipment"/>
<owl:disjointWith rdf:resource="#ModelCharacteristics"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Geometry</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Ground-Support_Equipment">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Ground-Support Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Hydraulic_System">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>

```

```

<owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
<owl:disjointWith rdf:resource="#Passenger_Equipment"/>
<owl:disjointWith rdf:resource="#Preparation_Air_System"/>
<owl:disjointWith rdf:resource="#Pressure_Control_System"/>
<owl:disjointWith rdf:resource="#Protection_Equipment"/>
<owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
<owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Hydraulic System</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Mass_Characteristics">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#uses"/>
      <owl:someValuesFrom rdf:resource="#Geometrical_Model"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#ModelCharacteristics"/>
  <owl:disjointWith rdf:resource="#Center_of_Mass"/>
  <owl:disjointWith rdf:resource="#Name_of_Model"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Mass Characteristic</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Master_Geometry">
  <rdfs:subClassOf rdf:resource="#Geometrical_Model"/>
  <owl:disjointWith rdf:resource="#Planer"/>
  <owl:disjointWith rdf:resource="#Power_Plant"/>
  <owl:disjointWith rdf:resource="#Systems_and_Equipment"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Master geometry</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Mid-Flight_Engine_Control">
  <rdfs:subClassOf rdf:resource="#Power_Plant"/>
  <owl:disjointWith rdf:resource="#Auxiliary_Power_Plant"/>
  <owl:disjointWith rdf:resource="#Fire_Protection_System"/>
  <owl:disjointWith rdf:resource="#Fuel_System"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Installation"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Nacelle"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Mid-Flight Engine Control</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Mid-Flight_Engine_Installation">
  <rdfs:subClassOf rdf:resource="#Power_Plant"/>
  <owl:disjointWith rdf:resource="#Auxiliary_Power_Plant"/>
  <owl:disjointWith rdf:resource="#Fire_Protection_System"/>
  <owl:disjointWith rdf:resource="#Fuel_System"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Control"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Nacelle"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Mid-Flight Engine Installation</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Mid-Flight_Engine_Nacelle">
  <rdfs:subClassOf rdf:resource="#Power_Plant"/>
  <owl:disjointWith rdf:resource="#Auxiliary_Power_Plant"/>
  <owl:disjointWith rdf:resource="#Fire_Protection_System"/>
  <owl:disjointWith rdf:resource="#Fuel_System"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Control"/>
  <owl:disjointWith rdf:resource="#Mid-Flight_Engine_Installation"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Mid-Flight Engine Nacelle</rdfs:label>

```

```

</owl:Class>
<owl:Class rdf:ID="ModelCharacteristics">
  <owl:disjointWith rdf:resource="#Documentation"/>
  <owl:disjointWith rdf:resource="#Equipment"/>
  <owl:disjointWith rdf:resource="#Geometrical_Model"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Attribute</rdfs:label>
</owl:Class>
<ModelCharacteristics rdf:ID="ModelCharacteristics_1"/>
<owl:DatatypeProperty rdf:ID="name">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Documentation"/>
        <owl:Class rdf:about="#Equipment"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
        <owl:Class rdf:about="#ModelCharacteristics"/>
        <owl:Class rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">name</rdfs:label>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:Class rdf:ID="Name_of_Model">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#uses"/>
      <owl:someValuesFrom rdf:resource="#Design_Recommendations"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#ModelCharacteristics"/>
  <owl:disjointWith rdf:resource="#Center_of_Mass"/>
  <owl:disjointWith rdf:resource="#Mass_Characteristics"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Name Of Model</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="On-Board_Recorder">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >On-Board Recorder</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Oxygen_Equipment">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>

```



```

<owl:disjointWith rdf:resource="#Air_Condensation_System"/>
<owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
<owl:disjointWith rdf:resource="#Anti-Ice_System"/>
<owl:disjointWith rdf:resource="#Control_System"/>
<owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
<owl:disjointWith rdf:resource="#Freight_Equipment"/>
<owl:disjointWith rdf:resource="#Furnishings"/>
<owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
<owl:disjointWith rdf:resource="#Hydraulic_System"/>
<owl:disjointWith rdf:resource="#On-Board_Recorder"/>
<owl:disjointWith rdf:resource="#Passenger_Equipment"/>
<owl:disjointWith rdf:resource="#Preparation_Air_System"/>
<owl:disjointWith rdf:resource="#Pressure_Control_System"/>
<owl:disjointWith rdf:resource="#Protection_Equipment"/>
<owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
<owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Oxygen Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Passenger_Equipment">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Passenger Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Planer">
  <rdfs:subClassOf rdf:resource="#Geometrical_Model"/>
  <owl:disjointWith rdf:resource="#Master_Geometry"/>
  <owl:disjointWith rdf:resource="#Power_Plant"/>
  <owl:disjointWith rdf:resource="#Systems_and_Equipment"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Planer</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Power_Plant">
  <rdfs:subClassOf rdf:resource="#Geometrical_Model"/>
  <owl:disjointWith rdf:resource="#Master_Geometry"/>
  <owl:disjointWith rdf:resource="#Planer"/>
  <owl:disjointWith rdf:resource="#Systems_and_Equipment"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Power plant</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Preparation_Air_System">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>

```

```

<owl:disjointWith rdf:resource="#Control_System"/>
<owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
<owl:disjointWith rdf:resource="#Freight_Equipment"/>
<owl:disjointWith rdf:resource="#Furnishings"/>
<owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
<owl:disjointWith rdf:resource="#Hydraulic_System"/>
<owl:disjointWith rdf:resource="#On-Board_Recorder"/>
<owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
<owl:disjointWith rdf:resource="#Passenger_Equipment"/>
<owl:disjointWith rdf:resource="#Pressure_Control_System"/>
<owl:disjointWith rdf:resource="#Protection_Equipment"/>
<owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
<owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Preparation Air System</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Pressure_Control_System">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Pressure Control System</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Protection_Equipment">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Protection Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Pylon">

```

```

<rdfs:subClassOf rdf:resource="#Planer"/>
<owl:disjointWith rdf:resource="#Empennage"/>
<owl:disjointWith rdf:resource="#Fuselage"/>
<owl:disjointWith rdf:resource="#Undercarriage"/>
<owl:disjointWith rdf:resource="#Wing"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Pylon</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Radio_Communication_Equipment">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>
  <owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
  <owl:disjointWith rdf:resource="#Hydraulic_System"/>
  <owl:disjointWith rdf:resource="#On-Board_Recorder"/>
  <owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
  <owl:disjointWith rdf:resource="#Passenger_Equipment"/>
  <owl:disjointWith rdf:resource="#Preparation_Air_System"/>
  <owl:disjointWith rdf:resource="#Pressure_Control_System"/>
  <owl:disjointWith rdf:resource="#Protection_Equipment"/>
  <owl:disjointWith rdf:resource="#Water-Vacuum_System"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Radio Communication Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Server_Equipment">
  <rdfs:subClassOf rdf:resource="#Equipment"/>
  <owl:disjointWith rdf:resource="#Design_Equipment"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Server Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Systems_and_Equipment">
  <rdfs:subClassOf rdf:resource="#Geometrical_Model"/>
  <owl:disjointWith rdf:resource="#Master_Geometry"/>
  <owl:disjointWith rdf:resource="#Planer"/>
  <owl:disjointWith rdf:resource="#Power_Plant"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Systems and Equipment</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Technical_Task">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#used"/>
      <owl:someValuesFrom rdf:resource="#Geometrical_Model"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Documentation"/>
  <owl:disjointWith rdf:resource="#Commoning_Scheme"/>
  <owl:disjointWith rdf:resource="#Data_for_Next_Stage"/>
  <owl:disjointWith rdf:resource="#Design_Recommendations"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Technical Task</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Undercarriage">
  <rdfs:subClassOf rdf:resource="#Planer"/>
  <owl:disjointWith rdf:resource="#Empennage"/>
  <owl:disjointWith rdf:resource="#Fuselage"/>

```

```

<owl:disjointWith rdf:resource="#Pylon"/>
<owl:disjointWith rdf:resource="#Wing"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Undercarriage</rdfs:label>
</owl:Class>
<owl:ObjectProperty rdf:ID="used">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Commoning_Scheme"/>
        <owl:Class rdf:about="#Design_Recommendations"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
        <owl:Class rdf:about="#Technical_Task"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <owl:inverseOf rdf:resource="#uses"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Center_of_Mass"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
        <owl:Class rdf:about="#Mass_Characteristics"/>
        <owl:Class rdf:about="#Name_of_Model"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="uses">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Center_of_Mass"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
        <owl:Class rdf:about="#Mass_Characteristics"/>
        <owl:Class rdf:about="#Name_of_Model"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <owl:inverseOf rdf:resource="#used"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Commoning_Scheme"/>
        <owl:Class rdf:about="#Design_Recommendations"/>
        <owl:Class rdf:about="#Geometrical_Model"/>
        <owl:Class rdf:about="#Technical_Task"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:range>
</owl:ObjectProperty>
<owl:Class rdf:ID="Water-Vacuum_System">
  <rdfs:subClassOf rdf:resource="#Systems_and_Equipment"/>
  <owl:disjointWith rdf:resource="#Air_Condensation_System"/>
  <owl:disjointWith rdf:resource="#Aircraft_Instrumentation"/>
  <owl:disjointWith rdf:resource="#Anti-Ice_System"/>
  <owl:disjointWith rdf:resource="#Control_System"/>
  <owl:disjointWith rdf:resource="#Electric_System_Ensuring"/>
  <owl:disjointWith rdf:resource="#Freight_Equipment"/>
  <owl:disjointWith rdf:resource="#Furnishings"/>

```

```

<owl:disjointWith rdf:resource="#Ground-Support_Equipment"/>
<owl:disjointWith rdf:resource="#Hydraulic_System"/>
<owl:disjointWith rdf:resource="#On-Board_Recorder"/>
<owl:disjointWith rdf:resource="#Oxygen_Equipment"/>
<owl:disjointWith rdf:resource="#Passenger_Equipment"/>
<owl:disjointWith rdf:resource="#Preparation_Air_System"/>
<owl:disjointWith rdf:resource="#Pressure_Control_System"/>
<owl:disjointWith rdf:resource="#Protection_Equipment"/>
<owl:disjointWith rdf:resource="#Radio_Communication_Equipment"/>
<rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Water-Vacuum System</rdfs:label>
</owl:Class>
<owl:Class rdf:ID="Wing">
  <rdfs:subClassOf rdf:resource="#Planer"/>
  <owl:disjointWith rdf:resource="#Empennage"/>
  <owl:disjointWith rdf:resource="#Fuselage"/>
  <owl:disjointWith rdf:resource="#Pylon"/>
  <owl:disjointWith rdf:resource="#Undercarriage"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Wing</rdfs:label>
</owl:Class>
</rdf:RDF>

```

4 Реалізація зв'язку знання-орієнтованого керування параметрами геометричних моделей за допомогою агентів, використовуючи JADE

```

package com.AD;
import jade.core.Agent;
import com.AD.ontomap.impl.*;
import java.util.LinkedList;
import java.util.List;

```

```

public class CheckAgent extends Agent {
    public static void main(String[] args) throws NoSuchMethodException {
        OntoInterfacelImpl oint = new OntoInterfacelImpl("onto/AD1_v2.owl");
        List oint_CD_L_max = new LinkedList();
        oint_CD_L_max.add(oint.getValue(cd_l));
        List oint_CD_ANGLE_max = new LinkedList();
        oint_CD_ANGLE_max.add(oint.getValue(cd_angle));
        List oint_RAMP_L_max = new LinkedList();
        oint_RAMP_L_max.add(oint.getValue(ramp_l));
        List oint_RAMP_ANGLE_max = new LinkedList();
        oint_RAMP_ANGLE_max.add(oint.getValue(ramp_angle));
        List oint_TRAP_L_max = new LinkedList();
        oint_TRAP_L_max.add(oint.getValue(trap_l));
        List oint_TRAP_ANGLE_max = new LinkedList();
        oint_TRAP_ANGLE_max.add(oint.getValue(trap_angle));
        List oint_DISTANCE_TO_CD_X_max = new LinkedList();
        oint_DISTANCE_TO_CD_X_max.add(oint.getValue(distance_to_cd_x));
        List oint_DISTANCE_TO_CD_Y_max = new LinkedList();
        oint_DISTANCE_TO_CD_Y_max.add(oint.getValue(distance_to_cd_y));
        Connection connection = null;

        PreparedStatement preparedStatement = null;
        java.sql.ResultSet resultset = null;
        DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
        connection = DriverManager.getConnection("jdbc:oracle:thin:@otp-server:1521:vk", "vk2",
"vk2");
    }
}

```

```

        connection.setAutoCommit(false);
        preparedStatement = connection.prepareStatement(
            "SELECT VARIANT, CD_L, CD_ANGLE, RAMP_L, RAMP_ANGLE, TRAP_L,
TRAP_ANGLE, DISTANCE_TO_CD_X, DISTANCE_TO_CD_Y FROM VK2.LG_PARAM ",
            ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
        resultset = preparedStatement.executeQuery();
        List VARIANT=new LinkedList();
        List CD_L=new LinkedList();
        List CD_ANGLE=new LinkedList();
        List RAMP_L=new LinkedList();
        List RAMP_ANGLE=new LinkedList();
        List TRAP_L=new LinkedList();
        List TRAP_ANGLE=new LinkedList();
        List DISTANCE_TO_CD_X=new LinkedList();
        List DISTANCE_TO_CD_Y=new LinkedList();
        while(resultset.next){
            VARIANT.add(resultset.getString(CD_L));
            CD_L.add(resultset.getString(CD_L));
            CD_ANGLE.add(resultset.getString(CD_ANGLE));
            RAMP_L.add(resultset.getString(RAMP_L));
            RAMP_ANGLE.add(resultset.getString(RAMP_ANGLE));
            TRAP_L.add(resultset.getString(TRAP_L));
            TRAP_ANGLE.add(resultset.getString(TRAP_ANGLE));
            DISTANCE_TO_CD_X.add(resultset.getString(DISTANCE_TO_CD_X));
            DISTANCE_TO_CD_Y.add(resultset.getString(DISTANCE_TO_CD_Y));
        }

```

```

String[] casual_array = { CD_L, CD_ANGLE, RAMP_L, RAMP_ANGLE, TRAP_L, TRAP_ANGLE, DISTANCE_TO_CD_X,
DISTANCE_TO_CD_Y};
String[] max_array={ oint_CD_L_max, oint_CD_ANGLE_max, oint_RAMP_L_max, oint_RAMP_ANGLE_max,
oint_TRAP_L_max, oint_TRAP_ANGLE_max, oint_DISTANCE_TO_CD_X_max, oint_DISTANCE_TO_CD_Y_max };
for(int i=0; i< casual_max.length; i++) {
    checkAgent.signalAgent(casual[i], max[i]);
}
}

```

```

public static void signalAgent(String[] casual_array, max_array)    {
    for(String max_temp: max_array) {
        int count = 0;
        for(String temp : casual_array) { count ++;
            if(casual_temp > max_temp) {
                System.out.println("Error: Value " + casual_temp + " is out of limit" + max_temp);
                preparedStatement = connection.prepareStatement(
                    "update VK2.LG_PARAM set " +temp+"="+max_temp+"
                    where VARIANT = "+VARIANT.get(count) + ",
ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
                connection.commit();
            }
        }
    }
}

```

Додаток В

Лістинг програм

Реалізація зв'язку знання-орієнтованого керування параметрами геометричних моделей за допомогою програмних модулів

```

package Agent;
import rampRotate.Form;
public class CheckInput {
    public boolean passFlag;
    public CheckInput(/*Diagnostic diagnostic,*/String RAMP_L, String RAMP_angle, String TRAP_L, String
TRAP_angle, String CD_L, String CD_angle, String distance_to_CD_X, String distance_to_CD_Y) throws Exception {
        LoadLimits limits = new LoadLimits();
        passFlag=true;
        double ramp_l=Double.parseDouble(RAMP_L);
        if( ramp_l>limits.RAMP_L[limits.max_flag] || ramp_l<limits.RAMP_L[limits.min_flag] ){
            Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
            diagnostic.setStatus("Довжина рампи не відповідає встановленим
обмеженням__("+limits.RAMP_L[limits.min_flag] +" )__"+ramp_l+"__("+limits.RAMP_L[limits.max_flag] +" )");
            passFlag=false;
        }
        double ramp_angle=Double.parseDouble(RAMP_angle);
        if( ramp_angle>limits.RAMP_angle[limits.max_flag] || ramp_angle<limits.RAMP_angle[limits.min_flag]
){
            Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
            diagnostic.setStatus("Кут рампи не відповідає встановленим
обмеженням__("+limits.RAMP_angle[limits.min_flag] +" )__"+ramp_angle+"__("+limits.RAMP_angle[limits.max_flag]
+" )");
            passFlag=false;
        }
        double trap_L=Double.parseDouble(TRAP_L);
        if( trap_L>limits.TRAP_L[limits.max_flag] || trap_L<limits.TRAP_L[limits.min_flag] ){
            Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
            diagnostic.setStatus("Довжина трапу не відповідає встановленим
обмеженням__("+limits.TRAP_L[limits.min_flag] +" )__"+trap_L+"__("+limits.TRAP_L[limits.max_flag] +" )");
            passFlag=false;
        }
        double trap_angle=Double.parseDouble(TRAP_angle);
        if( trap_angle>limits.TRAP_angle[limits.max_flag] || trap_angle<limits.TRAP_angle[limits.min_flag] ){
            Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
            diagnostic.setStatus("Кут трапу не відповідає встановленим
обмеженням__("+limits.TRAP_angle[limits.min_flag] +" )__"+trap_angle+"__("+limits.TRAP_angle[limits.max_flag] +" )");
            passFlag=false;
        }
        double cd_L=Double.parseDouble(CD_L);
        if( cd_L>limits.CD_L[limits.max_flag] || cd_L<limits.CD_L[limits.min_flag] ){
            Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
            diagnostic.setStatus("Довжина стулки не відповідає встановленим
обмеженням__("+limits.CD_L[limits.min_flag] +" )__"+cd_L+"__("+limits.CD_L[limits.max_flag] +" )");
            passFlag=false;
        }
        double cd_angle=Double.parseDouble(CD_angle);
        if( cd_angle>limits.CD_angle[limits.max_flag] || cd_angle<limits.CD_angle[limits.min_flag]
{
            Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
            diagnostic.setStatus("Кут стулки не відповідає встановленим
обмеженням__("+limits.CD_angle[limits.min_flag] +" )__"+cd_angle+"__("+limits.CD_angle[limits.max_flag] +" )");
            passFlag=false;
        }
    }
}

```

```

    }
    double dist_to_CD_X=Double.parseDouble(distance_to_CD_X);
    if( dist_to_CD_X>limits.distance_to_CD_X[limits.max_flag] ||
dist_to_CD_X<limits.distance_to_CD_X[limits.min_flag] ){
        Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
        diagnostic.setStatus("Відстань до точки обертання стулки(по X) не відповідає встановленим
обмеженням__("+limits.distance_to_CD_X[limits.min_flag]+" )__"+dist_to_CD_X+"__("+limits.distance_to_CD_X[limits.m
ax_flag]+" )");
        passFlag=false;
    }
    double dist_to_CD_Y=Double.parseDouble(distance_to_CD_Y);
    if( dist_to_CD_Y>limits.distance_to_CD_Y[limits.max_flag] ||
dist_to_CD_Y<limits.distance_to_CD_Y[limits.min_flag] ){
        Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
        diagnostic.setStatus("Відстань до точки обертання стулки(по Y) не відповідає встановленим
обмеженням__("+limits.distance_to_CD_Y[limits.min_flag]+" )__"+dist_to_CD_Y+"__("+limits.distance_to_CD_Y[limits.m
ax_flag]+" )");
        passFlag=false;
    }
    double CD_RAMP_check=ramp_l+cd_L;
    if( CD_RAMP_check>limits.CD_CSYS_Limit_max || CD_RAMP_check<limits.CD_CSYS_Limit_min){
        //diagnostic.setStatus("Сумарна довжина трапу та стулки__("+limits.CD_CSYS_Limit+" )__ не
співпадає з вхідними даними__("+CD_RAMP_check+" )");
        double newCD = limits.CD_CSYS_Limit_min-ramp_l;
        Diagnostic diagnostic =new Diagnostic("Коригуючий агент");
        if( newCD>limits.CD_L[limits.max_flag] || newCD<limits.CD_L[limits.min_flag] ){
            diagnostic.setStatus("Розрахована довжина стулки не відповідає встановленим
обмеженням__("+limits.CD_L[limits.min_flag] +" )__"+newCD+"__("+limits.CD_L[limits.max_flag] +" )");
            passFlag=false;System.out.println("false check ");
        }else{
            diagnostic.setStatus("Довжина стулки була змінена");
            Form.CD_L=String.format("%.5f", (newCD));
            Form.CD_L=Form.CD_L.substring(0,Form.CD_L.lastIndexOf(", "));
        }
    }
    if( passFlag){
        Diagnostic diagnostic =new Diagnostic("Перевіряючий агент");
        diagnostic.setStatus("Дані відповідають обмеженням");
    }
}
}
package Agent;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import fuzzy_logic.ConnectionManager;
import fuzzy_logic.RuleManager;
import fuzzy_logic.RuleType;
import fuzzy_logic.Rules;
public class CreateLimits {
    private ConnectionManager      cm;
    public List<Integer>           fuzzValues;
    public CreateLimits(ConnectionManager cm) throws SQLException {
        this.cm = cm;
        fuzzValues = new ArrayList<>();
    }
    public void loadLimits(RuleType rt) throws SQLException{
        double[] limits = makeLimits(rt);
        for (int i = 2; i >= 1; i--)

```



```

        {
            int value = (int) limits[i - 1];
            cm.setQuery("UPDATE `ramp`.`limits` SET `ramp_1`='" + value + "' WHERE `variant`='" + i + "'");
        }
    }
private double[] makeLimits(RuleType rt) throws SQLException{
    double[] limits = new double[2];
    double fuzzValue = 0.0;
    switch(rt)
    {
        case RAMP:
            fuzzValue = getRampFuzzySet();
            break;
    }
    fuzzValues.add((int)fuzzValue);
    limits[1] = fuzzValue * 0.9850;
    limits[0] = fuzzValue * 1.0150;
    return (limits);
}
private double getRampFuzzySet() throws SQLException{
    RuleManager rm = new RuleManager();
    rm.addNew(new Rules(cm, RuleType.RAMP.getList()));
    rm.fuzzification("and");
    rm.defuzzification();
    return (rm.result);
}
}
package Agent;
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.TextArea;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
public class Diagnostic {
    public JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JTextArea textArea = new JTextArea(100,100);
    public Diagnostic(String frameName){
        prepareGUI(frameName);
    }
    public void setStatus(String input){
        statusLabel.setText(input);
    }
    private void prepareGUI(String frameName){
        mainFrame = new JFrame(frameName);
        mainFrame.setSize(800,80);
        //mainFrame.setLayout(new GridLayout(3,1));
        mainFrame.addWindowListener(new WindowAdapter(){
        });
        headerLabel=new JLabel("",JLabel.CENTER);
        statusLabel=new JLabel("",JLabel.CENTER);
        statusLabel.setSize(110,350);
    }
}

```

```

        mainFrame.add(headerLabel);
        mainFrame.add(statusLabel);
        //mainFrame.add(scrollPane, BorderLayout.CENTER);
        mainFrame.setVisible(true);
        //setStatus("Агент готовий");
    }
    /**-----**/
    public void InfoMass(String frameName){
        mainFrame = new JFrame(frameName);
        mainFrame.setSize(500,500);
        //mainFrame.setLayout(new GridLayout(3,1));
        mainFrame.addWindowListener(new WindowAdapter(){
        });
        JScrollPane scrollPane = new JScrollPane(textArea);

        mainFrame.add(scrollPane, BorderLayout.CENTER);
        mainFrame.setVisible(true);
    }
    public void setInfo(String input){
        textArea.setLineWrap(true);
        textArea.append(input);
    }
    /**-----**/
}
package Agent;
import rampRotate.DBupdate;
public class LoadLimits {
    protected double[] RAMP_L = new double[2];
    protected double[] RAMP_angle = new double[2];
    protected double[] TRAP_L = new double[2];
    protected double[] TRAP_angle = new double[2];
    protected double[] CD_L = new double[2];
    protected double[] CD_angle = new double[2];
    protected double[] distance_to_CD_X = new double[2];
    protected double[] distance_to_CD_Y = new double[2];
    protected double CD_CSYS_Limit_max;
    protected double CD_CSYS_Limit_min;
    protected int max_flag;
    protected int min_flag;
    protected String[] positionNaim = new String[2];
    protected String[] var = new String[2];
    protected LoadLimits() throws Exception {
        byte flag = 1;
        DBupdate setConnection = new DBupdate();
        setConnection.connectionSetings("VK2.KB_CD");
        java.sql.ResultSet localResultSet = setConnection.selectSetings(flag, "178");
        int counter = 0;
        while (localResultSet.next()) {
            this.RAMP_L[counter] = Double.parseDouble(localResultSet.getString(1));
            this.RAMP_angle[counter] = Double.parseDouble(localResultSet.getString(2));
            this.TRAP_L[counter] = Double.parseDouble(localResultSet.getString(3));
            this.TRAP_angle[counter] = Double.parseDouble(localResultSet.getString(4));
            this.CD_L[counter] = Double.parseDouble(localResultSet.getString(5));
            this.CD_angle[counter] = Double.parseDouble(localResultSet.getString(6));
            this.distance_to_CD_X[counter] = Double.parseDouble(localResultSet.getString(7));
            this.distance_to_CD_Y[counter] = Double.parseDouble(localResultSet.getString(8));
            this.var[counter] = localResultSet.getString(9);
            this.positionNaim[counter] = localResultSet.getString(10);
            counter++;
        }
    }
}

```

```

    }
    for (int i = 0; i < this.positionNaim.length; i++) {
        if (this.positionNaim[i].equals("max")) {
            this.max_flag = i;
            this.CD_CSYS_Limit_max = Math.sqrt((this.distance_to_CD_X[i] *
this.distance_to_CD_X[i]) + (this.distance_to_CD_Y[i] * this.distance_to_CD_Y[i]));
        } else {
            this.min_flag = i;
            this.CD_CSYS_Limit_min = Math.sqrt((this.distance_to_CD_X[i] *
this.distance_to_CD_X[i]) + (this.distance_to_CD_Y[i] * this.distance_to_CD_Y[i]));
        }
    }
}

package fuzzy_logic;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class ConnectionManager {
    private Connection          connection;
    private PreparedStatement    preparedStatement;
    private ResultSet           resultSet;
    private String              table;
    public ConnectionManager(String addr,String log, String pass) throws SQLException {
        connection = null;
        preparedStatement = null;
        resultSet = null;
        if (addr.contains("oracle"))
            DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
        else if (addr.contains("mysql"))
            DriverManager.registerDriver(new com.mysql.jdbc.Driver());
        this.connection = DriverManager.getConnection(addr, log, pass);
        this.connection.setAutoCommit(false);
    }
    public void setQuery(String query) throws SQLException
    {
        preparedStatement = connection.prepareStatement(query, ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);
        if (query.toLowerCase().contains("select"))
            resultSet = preparedStatement.executeQuery();
        else
        {
            preparedStatement.execute();
            connection.commit();
        }
    }
    public void terminate() throws SQLException
    {
        preparedStatement.close();
        resultSet.close();
        connection.close();
    }
    public ResultSet getResultset() {
        return resultSet;
    }
    public void setResultset(ResultSet resultSet) {
        this.resultset = resultSet;
    }

```

```

    }
    public void setTable(String table) {
        this.table = table;
    }
    public String getTable() {
        return table;
    }
}
package fuzzy_logic;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Map.Entry;
public class RuleManager {
    private List<Rules>                rules;
    private Map<Integer, List<Double>> fuzzificationSet;
    private Map<Integer, List<Double>> volumeSet;
    public Double                      result;
    public RuleManager() {
        rules = new ArrayList<Rules>();
        fuzzificationSet = new HashMap<Integer, List<Double>>();
        volumeSet = new HashMap<Integer, List<Double>>();
    }
    public void defuzzification()
    {
        for (Entry<Integer, List<Double>> entry : fuzzificationSet.entrySet())
        {
            List<Double> vol = new ArrayList<>(volumeSet.get(entry.getKey()));
            double integral1 = 0.0;
            for (double v : vol)
                integral1 += v;
            double integral2 = 0.0;
            for (double g : entry.getValue())
                integral2 += g;
            result = 1e6 * (integral1 / integral2) / 42.9;
        }
    }
    public void aggregation (String type)
    {
        int i = 0;
        for (Rules r : rules)
        {
            List<Double> fuzz = new ArrayList<>();
            List<Double> vol = new ArrayList<>();

            for (Entry<Integer, Integer> e : r.id.entrySet())
            {
                double ex = 0.0;
                if (type == "and")
                    ex = Math.max(r.estSet1.get(e.getKey() - 1), r.estSet2.get(e.getValue() - 1));
                else if (type == "or")
                    ex = Math.min(r.estSet1.get(e.getKey() - 1), r.estSet2.get(e.getValue() - 1));
                fuzz.add(ex);
                vol.add(ex * r.dataSet1.get(e.getKey() - 1) / r.dataSet2.get(e.getValue() - 1));
            }
            fuzzificationSet.put(i, fuzz);
            volumeSet.put(i, vol);
            i++;
        }
    }
}

```

```

    }
}
public void addNew(Rules r)
{
    if (r == null)
        return ;
    rules.add(r);
}
}
package fuzzy_logic;
import java.util.ArrayList;
import java.util.List;
public enum RuleType {
    RAMP
    {
        {
            this.list.add("ramp_rules");
            this.list.add("ramp_mass_conclusion");
            this.list.add("ramp_density_conclusion");
        }
    },
    TRAP
    {
        {
            this.list.add("trap_rules");
            this.list.add("trap_mass_conclusion");
            this.list.add("trap_density_conclusion");
        }
    },
    CD
    {
        {
            this.list.add("cd_rules");
            this.list.add("cd_mass_conclusion");
            this.list.add("cd_density_conclusion");
        }
    }
};
List<String> list;

public List<String> getList(){
    return (list);
}
private RuleType(){
    this.list = new ArrayList<String>();
}
}
package fuzzy_logic;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.Map.Entry;
public class test {
    public static void main(String[] args) throws SQLException {ConnectionManager cm = new
ConnectionManager("jdbc:mysql://localhost:3306/ramp", "root", "1111");
        RuleManager rm = new RuleManager();
        rm.addNew(new Rules(cm, RuleType.RAMP.getList()));
        rm.fuzzification("and");
        rm.defuzzification();
        System.out.println(rm.result);
    }
}

```

```

    }
}

```

Код забезпечення зв'язку між САх системою САТІА v5 та БД

```

imports MySql.Data.MySqlClient
Name
type Scope Value
myDb->
String-> User ->DataBaseName
myServer->
String-> User ->Servername
RAMP_L->
String-> User ->DB_RAMP_L
RAMP_angle->
String-> User ->DB_RAMP_angle
TRAP_L->
String-> User ->DB_TRAP_L
TRAP_angle->
String-> User ->DB_TRAP_angle
CD_L->
String-> User ->DB_CD_L
CD_angle->
String-> User ->DB_CD_angle
distance_to_CD_X->
String-> User ->DB_distance_to_CD_X
distance_to_CD_Y->
String-> User ->DB_distance_to_CD_Y
myPort->
String-> User ->mySQLPort(3306)
Module mdlDataConn public conn As New MySqlConnection
Public Sub ConnectionDatabase()
Try
if
conn.State=ConnectionState.Closed
Then
conn.ConnectionString="DATABASE" & My.Settings.myDB&amp;amp;"
&"SERVER="&My.Settings.myServer

&" RAMP_L="&MySettins.myRAMP_L_VALUE
&" RAMP_angle="&MySettins.myRAMP_angle_VALUE
&" TRAP_L="&MySettins.myTRAP_L_VALUE
&" TRAP_angle="&MySettins.myTRAP_angle_VALUE
&" CD_L="&MySettins.myCD_L_VALUE
&" CD_angle="&MySettins.myCD_angle_VALUE
&" distance_to_CD_X="&MySettins.mydistance_to_CD_X_VALUE
&" distance_to_CD_Y="&MySettins.my distance_to_CD_Y_VALUE

&My.Settings.myPort&amp;amp;" ;charset=utf8"

conn.Open()
End if
Catch
myerror As Exception MessageBox.Show("Error Connection to database",MessageBoxButtons.OK)
End
End try
End Sub
Public Sub

```

```

DisconnectDatabase()
try
conn.Close
Catch
myerror As MySql.Data.MySqlClient.MySqlExeption
End Try
End Sub
End Module
.....

ConnectDatabase
""""""DisconnectDatabase

Dim axisSystem1 As AxisSystem
Set axisSystem1 = axisSystems1.Add()
axisSystem1.OriginType = catAxisSystemOriginByCoordinates
Dim arrayOfVariantOfDouble1(2)
arrayOfVariantOfDouble1(0) = 0.000000
arrayOfVariantOfDouble1(1) = 0.000000
arrayOfVariantOfDouble1(2) = 0.000000
axisSystem1.PutOrigin arrayOfVariantOfDouble1
axisSystem1.XAxisType = catAxisSystemAxisByCoordinates
Dim arrayOfVariantOfDouble2(2)
arrayOfVariantOfDouble2(0) = 1.000000
arrayOfVariantOfDouble2(1) = 0.000000
arrayOfVariantOfDouble2(2) = 0.000000
axisSystem1.PutXAxis arrayOfVariantOfDouble2
axisSystem1.YAxisType = catAxisSystemAxisByCoordinates
Dim arrayOfVariantOfDouble3(2)
arrayOfVariantOfDouble3(0) = 0.000000
arrayOfVariantOfDouble3(1) = 1.000000
arrayOfVariantOfDouble3(2) = 0.000000
axisSystem1.PutYAxis arrayOfVariantOfDouble3
axisSystem1.ZAxisType = catAxisSystemAxisByCoordinates
Dim arrayOfVariantOfDouble4(2)
arrayOfVariantOfDouble4(0) = 0.000000
arrayOfVariantOfDouble4(1) = 0.000000
arrayOfVariantOfDouble4(2) = 1.000000
axisSystem1.PutZAxis arrayOfVariantOfDouble4
part1.UpdateObject axisSystem1
axisSystem1.IsCurrent = True
part1.Update
Dim settingControllers1 As SettingControllers
Set settingControllers1 = CATIA.SettingControllers
Dim visualizationSettingAtt1 As SettingController
Set visualizationSettingAtt1 = settingControllers1.Item("CATVizVisualizationSettingCtrl")
visualizationSettingAtt1.SaveRepository
Dim axisSystem2 As AxisSystem
Set axisSystem2 = axisSystems1.Add()
Dim hybridShapeFactory1 As Factory
Set hybridShapeFactory1 = part1.HybridShapeFactory
Dim hybridShapePointCoord1 As HybridShapePointCoord
Set hybridShapePointCoord1 = hybridShapeFactory1.AddNewPointCoord(DISTANCE_TO_CD_X, DISTANCE_TO_CD_Y,
0.000000)

Dim bodies1 As Bodies
Set bodies1 = part1.Bodies

Dim body1 As Body
Set body1 = bodies1.Item("PartBody")

```

```

body1.InsertHybridShape hybridShapePointCoord1

part1.InWorkObject = hybridShapePointCoord1

part1.Update

axisSystem2.OriginType = catAxisSystemOriginByPoint

Dim reference1 As Reference
Set reference1 = part1.CreateReferenceFromObject(hybridShapePointCoord1)

axisSystem2.OriginPoint = reference1

axisSystem2.XAxisType = catAxisSystemAxisSameDirection

Dim reference2 As Reference
Set reference2 =
part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(AxisSystem.1;1);None:());Cf11:());Face:(Brp:(AxisSystem.1;3);None:());Cf11:());None:(Limits1:();Limits2:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem1)

axisSystem2.XAxisDirection = reference2

axisSystem2.YAxisType = catAxisSystemAxisSameDirection

Dim reference3 As Reference
Set reference3 =
part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(AxisSystem.1;2);None:());Cf11:());Face:(Brp:(AxisSystem.1;1);None:());Cf11:());None:(Limits1:();Limits2:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem1)

axisSystem2.YAxisDirection = reference3

axisSystem2.ZAxisType = catAxisSystemAxisSameDirection

Dim reference4 As Reference
Set reference4 =
part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(AxisSystem.1;3);None:());Cf11:());Face:(Brp:(AxisSystem.1;2);None:());Cf11:());None:(Limits1:();Limits2:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem1)

axisSystem2.ZAxisDirection = reference4

part1.UpdateObject axisSystem2

axisSystem2.IsCurrent = True

part1.Update

visualizationSettingAtt1.SaveRepository

Dim specsAndGeomWindow1 As Window
Set specsAndGeomWindow1 = CATIA.ActiveWindow

Dim viewer3D1 As Viewer
Set viewer3D1 = specsAndGeomWindow1.ActiveViewer

viewer3D1.Reframe

```


Dim viewpoint3D1 As Viewpoint3D
Set viewpoint3D1 = viewer3D1.Viewpoint3D

Dim reference5 As Reference
Set reference5 =
part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(AxisSystem.1;1);None:();Cf11:()));Face:(Brp:(AxisSystem.1;3);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem1)

Dim reference6 As Reference
Set reference6 =
part1.CreateReferenceFromBRepName("RSur:(Face:(Brp:(AxisSystem.1;1);None:();Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem1)

Dim reference7 As Reference
Set reference7 =
part1.CreateReferenceFromBRepName("FVertex:(Vertex:(Neighbours:(Face:(Brp:(AxisSystem.1;2);None:();Cf11:()));Face:(Brp:(AxisSystem.1;3);None:();Cf11:());Face:(Brp:(AxisSystem.1;1);None:();Cf11:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem1)

Dim hybridShapeLineAngle1 As HybridShapeLineAngle
Set hybridShapeLineAngle1 = hybridShapeFactory1.AddNewLineAngle(reference5, reference6, reference7, False, 0.000000, RAMP_L, RAMP_ANGLE, False)

body1.InsertHybridShape hybridShapeLineAngle1

part1.InWorkObject = hybridShapeLineAngle1

part1.Update

Dim reference8 As Reference
Set reference8 =
part1.CreateReferenceFromBRepName("REdge:(Edge:(Face:(Brp:(AxisSystem.2;1);None:();Cf11:()));Face:(Brp:(AxisSystem.2;3);None:();Cf11:());None:(Limits1:();Limits2:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem2)

Dim reference9 As Reference
Set reference9 =
part1.CreateReferenceFromBRepName("RSur:(Face:(Brp:(AxisSystem.2;1);None:();Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem2)

Dim reference10 As Reference
Set reference10 = part1.CreateReferenceFromObject(hybridShapePointCoord1)

Dim hybridShapeLineAngle2 As HybridShapeLineAngle
Set hybridShapeLineAngle2 = hybridShapeFactory1.AddNewLineAngle(reference8, reference9, reference10, False, 0.000000, CD_L, CD_ANGLE, False)

body1.InsertHybridShape hybridShapeLineAngle2

part1.InWorkObject = hybridShapeLineAngle2

part1.Update

Dim reference11 As Reference
Set reference11 = part1.CreateReferenceFromObject(hybridShapeLineAngle1)

Dim reference12 As Reference

```

Set reference12 =
part1.CreateReferenceFromBRepName("RSur:(Face:(Brp:(AxisSystem.1;1);None:());Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)", axisSystem1)

Dim reference13 As Reference
Set reference13 =
part1.CreateReferenceFromBRepName("BorderFVertex:(BEdge:(Brp:(GSMLine.1;2);None:(Limits1:());Limits2:());-1);Cf11:());WithPermanentBody;WithoutBuildError;WithSelectingFeatureSupport;MFBRepVersion_CXR15)",
hybridShapeLineAngle1)

Dim hybridShapeLineAngle3 As HybridShapeLineAngle
Set hybridShapeLineAngle3 = hybridShapeFactory1.AddNewLineAngle(reference11, reference12, reference13, False,
0.000000, TRAP_L, TRAP_ANGLE, False)

body1.InsertHybridShape hybridShapeLineAngle3

part1.InWorkObject = hybridShapeLineAngle3

part1.Update
part1.Update

End Sub

```

Код забезпечення зв'язку між САХ системою NX 8 та БД

```

package rampRotate;
import nxopen.*;
import oracle.*;
import java.sql.*;
import rampRotate.GeometrySetup;
import rampRotate.DBupdate;
public class Run {
    static byte WHEREsectionFLAG = 1;// '1'- line with filter; '0'- all lines, no filters
    static String geom_varianValue = "1";// number of line
    static String modelPath="D:\\1-Work\\1-PHPTH\\4-Projects\\NX_DB\\rotate.prt";// path to the *.prt
file
    /*****setConnection.insertSetings input*****/
    static String height="5.0";
    /*****setConnection.deleteSetings input*****/
    static String geom_variant="6";//line with this number will be deleted
    /*****

    public static void main(String[] args) throws Exception {
        SessionCreator.getSession(modelPath);
        DBupdate setConnection= new DBupdate();
        setConnection.connectionSetings();
        GeometrySetup geometrySetup= new GeometrySetup();
        geometrySetup.setGeometry(modelPath,WHEREsectionFLAG,geom_varianValue);
    }//public static void main
}//public class Run

package rampRotate;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
public class DBupdate {
    Connection connection = null;

```

```

PreparedStatement preparedStatement = null;
java.sql.ResultSet resultset = null;
// -----
// Настройки соединения с Oracle
// -----
public void connectionSetings() throws Exception {
    // String connectionAddress = "\"jdbc:oracle:thin:@otp-server:1521:vk\", \"vk2\", \"vk2\"";
    DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
    connection = DriverManager.getConnection("jdbc:oracle:thin:@otp-server:1521:vk", "vk2", "vk2");
    connection.setAutoCommit(false);
} // public void connectionSetings()
// -----
// Запись столбцов LG_PARAM в resultset
// -----
public ResultSet selectSetings(Byte WHEREsectionFLAG, String geom_varianValue) throws Exception {
    String WHEREsection[] = { "", " WHERE variant=" + geom_varianValue + " " };
    preparedStatement = connection.prepareStatement(
        "SELECT RAMP_L,RAMP_angle,TRAP_L,TRAP_angle,CD_L,CD_angle,distance_to_CD_X
,distance_to_CD_Y,VARIANT,NAME FROM VK2.LG_PARAM "+WHEREsection[WHEREsectionFLAG]+"",
        ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
    resultset = preparedStatement.executeQuery();
    System.out.println("DB connection was successful : "+WHEREsection[WHEREsectionFLAG]);
    return resultset;
} // public void deletetSetings()
// -----
// Добавление строки в LG_PARAM
// -----
public void insertSetings(String VARIANT, String NAME, String RAMP_L, String RAMP_angle, String
TRAP_L, String TRAP_angle, String CD_L, String CD_angle, String distance_to_CD_X, String distance_to_CD_Y)
throws Exception {
    // -----
    // счетчик variant
    // -----
    int variant = 0;
    preparedStatement = connection.prepareStatement("SELECT variant FROM VK2.LG_PARAM ORDER BY
variant",
        ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
    resultset = preparedStatement.executeQuery();
    while (resultset.next()) {
        variant = resultset.getInt(1);
    } // while (resultset.next())
    variant += 1;
    preparedStatement = connection.prepareStatement(
        " INSERT INTO vk2.LG_PARAM (VARIANT,NAME,
RAMP_L,RAMP_angle,TRAP_L,TRAP_angle,CD_L,CD_angle,distance_to_CD_X ,distance_to_CD_Y) "
        + "VALUES (" +VARIANT+", "+NAME+", "+RAMP_L+", "+RAMP_angle+", "+TRAP_L
+", "+CD_L+", "+CD_angle+", "+distance_to_CD_X+", "+distance_to_CD_Y+")",
        ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
    preparedStatement.execute();
    connection.commit();
    connection.close();
    System.out.println("DB connection was successful");
} // public void insertSetings()
// -----
// Удаление строки в LG_PARAM по variant
// -----
public void deleteSetings(String variant) throws Exception {
    preparedStatement = connection.prepareStatement(
/*>*/ "DELETE FROM vk2.LG_PARAM WHERE variant=" + variant + " ",

```

```

        ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
        preparedStatement.execute();
        connection.commit();
        connection.close();
        System.out.println("DB connection was successful");
    } // public void deletetSetings()
} // public class DBupdate

package rampRotate;
import nxopen.CartesianCoordinateSystem;
import nxopen.NXObject;
import nxopen.Part;
import nxopen.PartLoadStatus;
import nxopen.PartSaveStatus;
import nxopen.Point3d;
import nxopen.Session;
import nxopen.SessionFactory;
import nxopen.Unit;
import nxopen.Vector3d;
import nxopen.Xform;
import rampRotate.GeometrySetup;
public class GeometrySetup {
    static String RAMP_L;
    static String RAMP_angle;
    static String TRAP_L;
    static String TRAP_angle;
    static String CD_L;
    static String CD_angle;
    static String distance_to_CD_X;
    static String distance_to_CD_Y;
    static String positionNaim;
    public void setGeometry(String modelPath, byte WHEREsectionFLAG, String geom_varianValue) throws
Exception {
        if (WHEREsectionFLAG != 0) {
            DBupdate setConnection = new DBupdate();
            setConnection.connectionSetings();
            setConnection.selectSetings(WHEREsectionFLAG, geom_varianValue);
            java.sql.ResultSet localResultSet = setConnection.selectSetings(WHEREsectionFLAG,
geom_varianValue);
            while (localResultSet.next()) {
                RAMP_L = localResultSet.getString(1);
                RAMP_angle = localResultSet.getString(2);
                TRAP_L = localResultSet.getString(3);
                TRAP_angle = localResultSet.getString(4);
                CD_L = localResultSet.getString(5);
                CD_angle = localResultSet.getString(6);
                distance_to_CD_X = localResultSet.getString(7);
                distance_to_CD_Y = localResultSet.getString(8);

                positionNaim=localResultSet.getString(10);
            } // while (localResultSet.next())
            if (RAMP_L != null) {
                // -----
                // Вызов метода CD_base.setCD_Builder() для
                // СК и оси для створки
                // -----
                String CSYSName="1"; //номер КС
                CDbase CD_base=new
CDbase(); CD_base.setCD_Builder(CSYSName,distance_to_CD_X,distance_to_CD_Y);

```

```

// -----
// Вызов метода lineGeometry.getLineBuilder() для
// изменения параметров линий
// -----
String lineNames[]={"2","3","7","6"};//номер линии 2-CD|0,3-RAMP|1,7-TRAP|2,6-add|3
/* > */
LineGeometry lineGeometry2_CD=new
LineGeometry();lineGeometry2_CD.setLineBuilder(lineNames[0],CD_L,CD_angle);
/* > */
LineGeometry lineGeometry3_RAMP=new
LineGeometry();lineGeometry3_RAMP.setLineBuilder(lineNames[1],RAMP_L,RAMP_angle);
LineGeometry lineGeometry7_TRAP=new
LineGeometry();lineGeometry7_TRAP.setLineBuilder(lineNames[2],TRAP_L,TRAP_angle);
TrapPosition trapPosition=new TrapPosition();
if( positionNaim.equals("cd_landing")) {
trapPosition.setTrapPosition(lineNames[3],lineNames[2]);
}
else
{
trapPosition.setTrapPosition(lineNames[1],lineNames[2]);
}
System.out.println("Trap start point changed : " + positionNaim);
// -----
// Сохранить файл
// -----
PartSaveStatus partSaveStatus1;
partSaveStatus1 =
SessionCreator.getWorkPart().save(nxopen.BasePart.SaveComponents.TRUE,nxopen.BasePart.CloseAfterSave.FALSE);
partSaveStatus1.dispose();
partSaveStatus1 = null;
// System.out.println("GeometrySetup loaded || : " + " ; h/w/r
// : " + "/" + "/" + );
} else {
System.out.println("DB connection FAILURE");
} // if(height!=null)
} else {
System.out.println("More then one line had been choosen for ramp geometry");
} // if(WHEREsectionFLAG!=1){
} // public void getGeometrySetup(...)
} // public class GeometrySetup

package rampRotate;
import nxopen.Expression;
import nxopen.NXObject;
import nxopen.PartLoadStatus;
import nxopen.Plane;
import nxopen.Point3d;
import nxopen.Vector3d;
public class LineGeometry {
public void setLineBuilder(String lineName,String lineLength,String lineAngle) throws Exception {
// -----
// Выбор отрезка
// -----
nxopen.features.AssociativeLine associativeLine3 =
((nxopen.features.AssociativeLine)SessionCreator.getWorkPart().features().findObject("LINE("+lineName+"")));//19
// -----
// Конструктор линий
// -----
associativeLine3.makeCurrentFeature();
nxopen.features.AssociativeLineBuilder associativeLineBuilder2;

```

```

        associativeLineBuilder2 =
SessionCreator.getWorkPart().baseFeatures().createAssociativeLineBuilder(associativeLine3);
        // -----
        // Базовая точка отрезка , вектор направления
        // -----
        Point3d origin3 = new Point3d(0.0, 0.0, 0.0);
        Vector3d normal2 = new Vector3d(0.0, 0.0, 1.0);
        Plane plane2;
        plane2 = SessionCreator.getWorkPart().planes().createPlane(origin3, normal2,
nxopen.SmartObject.UpdateOption.WITHIN_MODELING);
        // -----
        // Координаты для конечной точки отрезка lineLength
        // Угол lineAngle
        // -----
        associativeLineBuilder2.limits().startLimit().distance().setRightHandSide("0");
/* > */ associativeLineBuilder2.limits().endLimit().distance().setRightHandSide(lineLength);//-6850
/* > */ associativeLineBuilder2.angle().setRightHandSide(lineAngle);//19.20
        NXObject nXObject3;
        nXObject3 = associativeLineBuilder2.commit();
        associativeLineBuilder2.destroy();
        System.out.println("Line_Builder loaded : "+lineLength+" : "+lineAngle);
    }
}

```

```

package rampRotate;
import nxopen.Part;
import nxopen.Session;
import nxopen.SessionFactory;
public class SessionCreator {
    public static Session theSession;
    public static Part workPart;
    public static Part displayPart;
    public static void getSession(String modelPath) throws Exception{
        theSession = (Session) SessionFactory.get("Session");
        nxopen.PartCollection.OpenBaseData openBaseData1;
        openBaseData1 = theSession.parts().openBaseDisplay(modelPath);
        openBaseData1.loadStatus.dispose();
        openBaseData1.loadStatus = null;
    }
    public static Session getTheSession() throws Exception{
        return theSession;}
    public static Part getDisplayPart() throws Exception{
        displayPart = theSession.parts().display();
        return displayPart;}
    public static Part getWorkPart() throws Exception{
        workPart = theSession.parts().work();
        return workPart;}
}

```

```

package rampRotate;
import nxopen.Point;
import nxopen.Point3d;
import nxopen.Line;
import nxopen.MeasureDistance;
import nxopen.NXObject;
import nxopen.Plane;
import nxopen.Scalar;
import nxopen.TaggedObject;
import nxopen.Unit;

```

```

import nxopen.Vector3d;
import nxopen.features.AssociativeLine;
public class TrapPosition {
    public void setTrapPosition(String startPointLine,String trapN) throws Exception {
        nxopen.features.AssociativeLine associativeLine1 =
((nxopen.features.AssociativeLine)SessionCreator.getWorkPart().features().findObject("LINE("+trapN+""));
        SessionCreator.getWorkPart().features().setEditWithRollbackFeature(associativeLine1);
        associativeLine1.makeCurrentFeature();
        nxopen.features.AssociativeLineBuilder associativeLineBuilder1;
        associativeLineBuilder1 = SessionCreator.getWorkPart().baseFeatures().createAssociativeLineBuilder(associativeLine1);
        Point3d origin1 = new Point3d(0.0, 0.0, 0.0);
        Vector3d normal1 = new Vector3d(0.0, 0.0, 1.0);
        Plane plane1;
        plane1 = SessionCreator.getWorkPart().planes().createPlane(origin1, normal1,
nxopen.SmartObject.UpdateOption.WITHIN_MODELING);
        Unit unit1;
        unit1 = associativeLineBuilder1.limits().startLimit().distance().units();
        // -----
        // startPoint().setValue(null);
        // -----
        associativeLineBuilder1.startPoint().setValue(null);
        Scalar scalar1;
        scalar1 = SessionCreator.getWorkPart().scalars().createScalar(1.0, nxopen.Scalar.DimensionalityType.NONE,
nxopen.SmartObject.UpdateOption.WITHIN_MODELING);
        nxopen.features.AssociativeLine associativeLine2 =
((nxopen.features.AssociativeLine)SessionCreator.getWorkPart().features().findObject("LINE("+startPointLine+""));// 6-
add or 3-ramp
        Line line1 = ((Line)associativeLine2.findObject("CURVE 1"));
        Point point1;
        point1 = SessionCreator.getWorkPart().points().createPoint(line1, scalar1,
nxopen.SmartObject.UpdateOption.WITHIN_MODELING);
        associativeLineBuilder1.limits().startLimit().distance().setRightHandSide("0");
        associativeLineBuilder1.startPoint().setValue(point1);
        associativeLineBuilder1.commit();
        TaggedObject taggedObject1;
        taggedObject1 = associativeLineBuilder1.startPoint().value();
        associativeLineBuilder1.destroy();
    }
}

```

Додаток Г
Акти впровадження

"ЗАТВЕРДЖУЮ"

Віце-президент з проектування
Державного підприємства "Антонов", к.т.н.



Лось О. В.

АКТ

впровадження результатів
дисертаційної роботи Конотопа Дмитра Ігоровича

09 04 2019 р.

м. Київ

Складено комісією на Державному підприємстві (ДП) "Антонов" у складі:
Головного конструктора з міцності, к.т.н. Семенця О. І.,
Головного конструктора з перспективних розробок, к.т.н. Філя С. А.,
Заст. головного конструктора з комп'ютерних інформаційних технологій, к.т.н.
Абрамова Ю. В.

Комісія провела роботу по визначенню фактичного впровадження результатів дисертаційної роботи інженера-конструктора 1 категорії Державного підприємства "Антонов" Конотопа Д. І. "Інформаційна технологія створення та супроводження узагальненої моделі складних технічних об'єктів" у виробничий процес, встановила наступне:

➤ в роботі досліджено та вперше сформульовано узагальнену модель складних технічних об'єктів, яка дозволяє вирішити задачу зв'язку моделей різних стадій життєвого циклу складних технічних об'єктів, отриманих з різних інформаційних систем. Вперше розроблено метод знання-орієнтованого створення складних технічних об'єктів і інформаційну технологію створення та супроводження узагальненої моделі складних технічних об'єктів. Розроблені модель, метод та інформаційна технологія дозволяють підвищити ефективність процесу створення літального апарату.

➤ розроблено засоби забезпечення реалізації інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів:

- базу даних "Параметри моделей", що ведеться в MySQL за допомогою системи керування базами даних DbVisualizer;
- базу знань, використовуючи програми Protege 4.3 та DbVisualizer, в якій зберігаються обмеження на складові бази даних відповідно вимогам нормативних документів, що створює швидкі та якісні умови для внесення змін в процес проектування і виробництва

літальних апаратів за допомогою спеціально налаштованих дій програмних модулів, розроблених в середовищі Java;

- програмні модулі для зв'язку з базою даних "Параметри моделей": інформаційної системи CAx NX в середовищі Java; та інформаційної системи CATIA в середовищі Visual Basic.

➤ розроблено інформаційну систему створення та супроводження узагальненої моделі складних технічних об'єктів, в якій приєднано розроблені інструментальні засоби до наявних серверів ДП "Антонов", з яких отримуються дані, що використовується в процесі проектування середнього транспортного літака АН-178 та легкого транспортного літака АН-132.

➤ робота доповнює технологію паралельного проектування інформаційної системи керування життєвим циклом продукту (PLM) та вдосконалює принципи використання компонентів інформаційної технології неперервної інформаційної підтримки життєвого циклу продукту (CALS), які використовуються на ДП "Антонов", за рахунок введення рівня керування параметрами моделей літального апарату з використанням бази даних та знань для завдання створення моделей літального апарату на прикладі використання інформаційних систем PLM TeamCenter та автоматизованого проектування, аналізу та виробництва (CAD/CAM/CAE) NX та CATIA.

Доведено ефективність розроблених результатів дисертаційної роботи Коногопа Дмитра Ігоровича, зокрема: при формуванні значень параметрів геометричних моделей деталей та складальних одиниць вантажного люку базової контрольної структури ескізного проекту середнього транспортного літака АН-178 та легкого транспортного літака АН-132 відповідно значенням обмежень вагової моделі та значенням моделі міцності; і при супроводженні геометричної моделі в інформаційних системах PLM TeamCenter та CAD/CAM/CAE NX, що показало близько 22% економії матеріальних витрат.

Даний акт не є основою для фінансових претензій до підприємства.

Головний конструктор
з міцності, к.т.н.



Семенець О. І.

Головний конструктор
з перспективних розробок, к.т.н.



Філь С. А.

Заст. головного конструктора
з комп'ютерних інформаційних
технологій, к.т.н.



Абрамов Ю. В.

"ЗАТВЕРДЖУЮ"
Декан факультету авіаційних та космічних систем
Національного технічного університету України
«Київський політехнічний інститут
імені Ігоря Сікорського»,
д.т.н., проф.
Збруцький О. В.



АКТ
впровадження результатів
дисертаційної роботи Конотопа Дмитра Ігоровича

26 06. 2018 р.

м. Київ

Складено комісією у складі:

зав. кафедри автоматизації експериментальних досліджень КПІ ім. Ігоря Сікорського, д.т.н., проф. Туза Ю. М.,
доц. кафедри автоматизації експериментальних досліджень КПІ ім. Ігоря Сікорського, к.т.н. Богомазова С. А.,
доц. кафедри автоматизації експериментальних досліджень КПІ ім. Ігоря Сікорського, к.т.н. Добролюбової М. В.

Комісія провела роботу по визначенню фактичного впровадження результатів дисертаційної роботи Конотопа Д. І. "Інформаційна технологія створення та супроводження узагальненої моделі складних технічних об'єктів" у навчально-науковий процес, встановила наступне:

Результати роботи використовувалися в рамках держбюджетної теми 2358 "Розробка засад створення нано- та пікосупутників як космічних мікролабораторій" (R/K/ 0110U002593).

Одержані наукові дослідження викладені у навчальному посібнику: Зінченко В.П. Обчислювальні мережі: побудова та програмування: навч. посібник / В.П. Зінченко, С.В. Зінченко, І.В. Мірошніченко, Д.І. Конотоп, В. Резаї – К. : НТУУ «КПІ», 2011. – 240 с.

Основні положення дисертаційної роботи Конотопа Дмитра Ігоровича використовуються у навчально-науковому процесі кафедри автоматизації експериментальних досліджень факультету авіаційних та космічних систем та є прикладом зв'язку з виробництвом. Розроблений метод знання-орієнтованого моделювання складних технічних об'єктів та реалізація інформаційної технології створення та супроводження узагальненої моделі складних технічних об'єктів використовуються в навчальному процесі при викладанні курсів на кафедрі автоматизації експериментальних досліджень

Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», зокрема в курсах:

«Багаторівневі системи обробки інформації»: в лекцію 10 додано інформацію про інформаційні технології проектних досліджень складного технічного об'єкта (на базі основних компонентів технології CALS – IC PLM та CAx показано співвідношення класичних етапів створення СТО та моделювання СТО, використовуючи інформаційні системи технології CALS; розкрито наявний процес моделювання СТО, котрий базується на методі інтегрованого проектування і комп'ютерного моделювання та на основі технології паралельного проектування PLM); лекція 18 доповнена представленням онтологічної схеми знань, розробкою багатоагентних систем та прикладами побудови бази знань складних технічних об'єктів; у лабораторній роботі 3 наведено зв'язок моделей складного технічного об'єкта на прикладі скидання вантажу з літака;

«Обчислювальні мережі»: лекція 4 доповнена веденням локальних обчислювальних мереж (ОМ) на підприємствах; в лекції 7 розкрито передачу даних в локальних ОМ; в лекції 10 представлено приклади ведення програм в локальних ОМ; у лекції 18 приведено вимоги до ОМ та методи їхнього проектування та адміністрування; у практичне заняття 5 додано способи і методи передачі даних та доступ до середовища обміну ОМ, наведено приклади протоколів керування каналами передачі даних; лабораторна робота 5 доповнена розробкою програм обміну даними клієнт-сервер з використанням протоколу IPX; лабораторна робота 6 доповнена розробкою програм обміну даними клієнт-сервер з використанням протоколу SPX та діагностики ОМ; в лабораторній роботі 7 розроблено та досліджено варіанти програмного забезпечення роботи з сервером з використанням можливостей ОМ.

Члени комісії:

Зав. кафедри автоматизації
експериментальних досліджень
НТУУ "КПІ" ім. Сікорського,
д.т.н., проф.



Туз Ю. М.

Доц. кафедри автоматизації
експериментальних досліджень,
НТУУ "КПІ" ім. Сікорського, к.т.н.



Богомазов С. А.

Доц. кафедри автоматизації
експериментальних досліджень
НТУУ "КПІ" ім. Сікорського, к.т.н.



Добролюбова М. В.

“ЗАТВЕРДЖУЮ”

В.о. ректора Національного університету
кораблебудування імені адмірала Макарова
Д-р техн. наук, професор

В.С. Блінцов



19. 01 2018 р.

про впровадження основних результатів дисертаційної роботи
«Інформаційна технологія створення узагальненої моделі складних
технічних об'єктів»
на здобуття наукового ступеня кандидата технічних наук
Конотопа Дмитра Ігоровича

Цей акт складено в тому, що основні положення та висновки дисертаційної роботи Конотопа Д. І. використовуються у навчально-науковому процесі Кораблебудівного навчально-наукового інституту Національного університету кораблебудування імені адмірала Макарова (КННІ НУК).

Розроблені в роботі інформаційна технологія, метод та узагальнена математична модель складних технічних об'єктів дозволяють вирішити задачу зв'язку моделей, створених в різноманітних інформаційних системах технології CALS, на різних стадіях життєвого циклу корабля.

Розроблений метод знання-орієнтованого моделювання складних технічних об'єктів та алгоритми використання методу: для керування параметричною інформацією при побудові моделей складних технічних об'єктів; для обміну даними між складовими CALS довели свою ефективність в процесі вирішення конкретних проектних завдань кораблебудування.

Результати роботи використовуються у навчально-науковому процесі при виконанні наукових досліджень, викладанні навчальних дисциплін та виконанні магістерських робіт зі спеціальності 135 «Суднобудування» на кафедрах «Теорія та проектування суден» та «Будівництва та ремонту суден» КННІ НУК.

Заступник директора КННІ НУК
з розвитку та інновацій,
академік Академії наук суднобудування України,
доктор технічних наук, професор

О. В. Щедролосоєв

Додаток Д

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Статті у наукових фахових виданнях

1. Зінченко В. П. Анализ средств и методов информационных технологий синтеза структур конечно-элементных моделей / В. П. Зінченко, В. В. Борисов, Д. І. Конотоп // Інформаційні системи, механіка та керування. – 2009. – № 3. – С. 112–121. *(автору належить: аналіз проблеми взаємодії моделей та IC PLM.)*
2. Зінченко В. П. Концепція застосування інтелектуальних технологій в проектуванні / В. П. Зінченко, Д. І. Конотоп, Г. В. Деркач, Є. Ю. Абрамов // Открытые информационные и компьютерные интегрированные технологии. – Харьков: Гос. Аэроком. Ун-т “ХАИ”, 2011. – № 49. – С. 169–179. *(автору належить: розкриття методу знання-орієнтованого створення складних технічних об'єктів та його модифікації.)*
3. Зінченко В. П. Інформаційні технології моделювання компоновки складного технічного об'єкта / В. П. Зінченко, Д. І. Конотоп, О. П. Сидоренко, В. В. Борисов // Інформаційні системи, механіка та керування. – 2011. – № 6. – С. 27–35. *(автору належить: формалізація проблематики розробки інформаційних технологій процесу побудови моделей.)*
4. Abramova A. V. Ontology application for estimation of complex technical object characteristics / A. V. Abramova, D. I. Konotop // Електроніка та системи управління. – 2012. – №3 (33). – С. 81–88. *(автору належить: опис методу та засобів знання-орієнтованого створення складних технічних об'єктів; його модифікацій.)*
5. Конотоп Д. І. Створення контрольної базової моделі в інформаційних технологіях проектування складного технічного об'єкта / Д. І. Конотоп, В. П. Зінченко // Наукові вісті Національного технічного університету України "Київський політехнічний інститут". – 2012. – № 6 (86). – С. 132–137. *(автору належить: розробка модифікації методу знання-орієнтованого створення складних технічних об'єктів для керування параметричною інформацією при створенні моделей.)*
6. Абрамова Г. В. Алгоритми планування вагових розрахунків літака / Г. В. Абрамова, Д. І. Конотоп // Вісник ¹⁷ Національного Авіаційного Університету. – 2013. – № 2 (55). – С. 129–135. *(автору належить: розробка методу та модифікації методу знання-орієнтованого створення складних технічних об'єктів.)*

Стаття у виданні, що входить до наукометричної бази Index Copernicus

7. Konotop D. I. 3D-models design concept of complex technical objects using knowledge-based technology/ D. I. Konotop, V. P. Zinchenko // Механіка гіроскопічних систем. – 2017. – № 34. – С. 5–13. *(автору належить: опис методу знання-орієнтованого створення складних технічних об'єктів та його модифікацій.)*

Стаття у іноземному виданні

8. Конотоп Д. И. Оптимальное проектирование сложных технических объектов с использованием онтологического подхода / Д. И. Конотоп, В. П. Зинченко // Научный журнал "Онтология проектирования". – Самара, "Новая техника". – 2011. – № 1(2). – С. 44–53. (*автору належить: опис методу знання-орієнтованого створення складних технічних об'єктів та його модифікацій.*)

Матеріали науково-технічних конференцій

9. Зінченко В. П. Проблеми оптимізації компоновки складного технічного об'єкту / В. П. Зінченко, С. В. Зінченко, Д. І. Конотоп, В. В. Борисов. // II наук. конференція магістрів та аспірантів, присвячена 20-річчю факультету прикладної математики (Київ, 14 - 16 квітня 2010 р.) – К.: НТУУ "КПІ" – 2010. – Тези доповідей. – С. 64–67.

10. Зінченко В. П. Застосування сучасних комп'ютерних інформаційних технологій при проектуванні складного технічного об'єкту / В. П. Зінченко, Д. І. Конотоп // Міжнародна науково-практична конференція "Інформаційні технології та комп'ютерна інженерія" (Вінниця, 19 - 21 травня 2010 р.) – Вінниця: ВНТУ – 2010. – Тези доповідей. – С. 50–51.

11. Konotop D. Multi-agent-based conception of modern aircraft design / D. Konotop, I. Budinska, V. Zinchenko, E. Gatial // Proceedings of 5th Workshop on Intelligent and Knowledge Oriented Technologies (Bratislava, Slovakia, November 11 - 12, 2010) – p. 125–128.

12. Абрамов Є. Ю. Алгоритмізація бази знань проектування складного технічного об'єкту / Є. Ю. Абрамов, Д. І. Конотоп, Г. В. Деркач // XVIII Міжнародна конференція з автоматичного управління "Автоматика/Automatics – 2011" (Львів, 28 - 30 вересня 2011 р.) – Львів: Видавництво Львівської політехніки – 2011 – Матеріали конференції. – С. 358–359.

13. Конотоп Д. І. Визначення керуючої моделі проектування складного технічного об'єкта / Д. І. Конотоп // XII Міжнародна науково-практична конференція молодих учених і студентів «Політ. Сучасні проблеми науки» (Київ, 4 - 5 квітня 2012 р.) – К.: НАУ – 2012. – Тези доповідей. – С. 69.

14. Конотоп Д. И. Использование базовой управляющей структуры в процессе проектирования сложного технического объекта / Д. И. Конотоп // XIX Международная научная конференция студентов, аспирантов и молодых ученых «ЛОМОНОСОВ-2012» (Москва, МГУ имени М. В. Ломоносова, 9 - 13 апреля 2012 г.) – М.: Издательский отдел факультета ВМК МГУ – 2012. – Сборник тезисов. – С. 29–30.

15. Abramova A. V. Application of Sheynin's plural calculations method at the initial stages of complex technical object design / A. V. Abramova, D. I. Konotop // 2-nd International Conference «Methods and Systems of Navigation and Motion Control» (Kyiv, October, 9 - 12, 2012) – К.: Освіта України – 2012. – Proceedings. – С. 129–132. (**Наукометрична база SCOPUS**)

16. Abramova A. V. Using graphs for the planning at the initial stages of complex technical object design / A. V. Abramova, D. I. Konotop // The 2nd International Scientific Conference of Students and Young Scientists "Theoretical and Applied

Aspects of Cybernetics" (Kyiv, November, 12 - 16, 2012) – К.: "Bukrek" – 2012. – Proceedings. – С. 8–13.

17. Конотоп Д. І. Застосування онтології в обміні даними при розробці складного технічного об'єкта / Д. І. Конотоп // Збірник доповідей ІХ Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки”. – Київ, 17 - 18 квітня 2013 – Секція "Інформаційні технології". – С. 437–440.

18. Abramov E. Knowledge-oriented support of complex technical object design / E. Abramov, D. Konotop, A. Abramova // 2-nd International Conference «Actual problems of UAV development» (Kyiv, October, 15 - 17, 2013) – Proceedings. – P. 122–125. (**Наукометрична база SCOPUS**)

19. Konotop D. Using ontology in geometrical modeling of complex technical object. / D. Konotop // VII International students and young scientists conference «Intelligence. Integration. Reliability» (Kyiv-Warsaw, April, 28 - 29, 2014) – К.: ІВЦ "Видавництво «Політехніка»" – 2014 – Abstracts. – P. 20–21.

20. Konotop D. Ontology using in geometrical models data processing of complex technical object / D. Konotop // XX-th International Conference «Knowledge-Dialogue-Solution» (Kyiv, September, 8 - 10, 2014) – ІТНЕА. Kyiv - Sofia – 2014 – Proceedings. – P. 118–119.

21. Конотоп Д. І. Використання знання-орієнтованих систем при створенні складного технічного об'єкта / Д. І. Конотоп, В. П. Зінченко, Лі Вей // Збірник доповідей Х Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки” (Київ, 16 - 17 квітня 2015) – Секція "Інформаційні технології". – С. 206–211.

22. Конотоп Д. І. Знання-орієнтована побудова геометричних моделей складних технічних об'єктів / Д. І. Конотоп, В. П. Зінченко // Збірник доповідей ХІ Міжнародної науково-технічної конференції “Гіротехнології, навігація, керування рухом і конструювання авіаційно-космічної техніки” (Київ, 13 - 14 квітня 2017) – Секція "Інформаційні технології". – С. 72–75.

23. Конотоп Д. І. Створення та супроводження узагальненої моделі складних технічних об'єктів / Д. І. Конотоп // Збірник доповідей ІХ Міжнародної науково-технічної конференції “Інновації в суднобудуванні та океанотехніці” (Миколаїв, 18 - 19 жовтня 2018) – Секція "Інформаційні технології та управління проектами в промисловості, освіті та соціальній сфері". – С. 402–405.

Навчальний посібник

24. Зінченко В. П. Обчислювальні мережі: побудова та програмування: навч. посібник / В. П. Зінченко, С. В. Зінченко, І. В. Мірошніченко, Д. І. Конотоп, В. Резаї – К. : НТУУ «КПІ», 2011. – 240 с.