

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

«На правах рукопису»

УДК _____

«До захисту допущено»

Завідувач кафедри

_____ Ткач М.М.
(підпис) (ініціали, прізвище)

“ ” _____ 2015 р.

Магістерська дисертація

зі спеціальності 8.05020102 Комп'ютеризовані та робототехнічні системи

на тему: “Побудова та дослідження інтегрованих систем автоматизованого проектування”

Виконав: студент VI курсу, групи ІК-31М

_____ Годний Андрій Павлович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник _____ к.т.н. доц. Чумаченко О. І. _____
(посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Консультант охорона праці _____ к.т.н. доц. Праховнік Н.А. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Консультант нормоконтроль _____ к.т.н. доц. Пасько В.П. _____
(посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____
(підпис)

Рецензент _____
(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____ (підпис)

Київ – 2015 року

Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

Освітньо-кваліфікаційний рівень «магістр»

Напрямок підготовки 6.050201 Системна інженерія

Спеціальність 8.05020102 Комп'ютеризовані та робототехнічні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри

Ткач М.М.

(підпис)

(ініціали, прізвище)

«__» _____ 2015 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Годного Андрія Павловича

(прізвище, ім'я, по батькові)

1. Тема дисертації: Побудова та дослідження інтегрованих систем
автоматизованого проектування

науковий керівник дисертації к.т.н., доц. Чумаченко О. І.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 10 » 03 2015 р. № 292/2С

2. Строк подання студентом дисертації 03.06.2015

3. Об'єкт дослідження процес автоматизованого проектування, виражений як
задача із структурним представленням даних

4. Предмет дослідження Модель сценарію автоматизованого проектування, що
забезпечує динамічну інтеграцію даних в єдиному інформаційному
просторі.

5. Перелік завдань, які потрібно розробити: дослідити існуючі системи
автоматизованого проектування, дослідити методи розробки складного
програмного забезпечення, розробити нове архітектурне рішення для побудови

інтегрованої системи автоматизованого проектування, перевірити його працездатність на різних даних.

6. Орієнтовний перелік ілюстративного матеріалу структурна схема інтегратора, структурна схема інтегрованої САПР, структура керуючого процесора, структура тематичного співпроцесора, структура бібліотеки операцій, схема керуючого процесора

7. Орієнтовний перелік публікацій Синєглазов В.М. Годний А. П. Динамічна інтеграція даних при проектуванні систем автоматизованого проектування // Науковий журнал. Електроніка та системи управління. – Київ: НАУ. 2014 – No. 2(40).С.51-58. Синєглазов В.М. Годний А. П. Структурний синтез інтегрованих систем автоматизованого проектування // Науковий журнал. Електроніка та системи управління. – Київ: НАУ. 2015 – No. 1(43).С.51-58.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
ОБ та БНС	к.т.н., доц. Праховник Н.А.		
Нормоконтроль	к.т.н., доц. Пасько В.П.		

9. Дата видачі завдання 21.10.2014

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Дослідження предметної області	22.10.2014	
2	Аналіз інтеграції даних і програмних засобів автоматизації	30.12.2014	
3	Дослідження методів побудови складних інтегрованих систем	10.04.2014	
4	Розробка інформаційної моделі середовища автоматизованого проектування.	23.05.2014	
5	Розробка методу динамічної інтеграції даних	18.11.2014	
6	Тестування системи	22.01.2015	
7	Оформлення документації по роботі	15.03.2015	
8	Оформлення ілюстрацій та схем	18.04.2015	
9	Здача дипломної роботи	03.06.2015	

Студент

(підпис)

А.П. Годний

(ініціали, прізвище)

Науковий керівник дисертації

(підпис)

О.І. Чумаченко

(ініціали, прізвище)

РЕФЕРАТ

Загальний обсяг магістерської дисертації складає 153 сторінки, що містить 10 таблиць, 7 плакатів, 24 ілюстрації, а також 71 бібліографічних найменувань за переліком посилань.

Робота присвячена питанням побудови інтегрованих систем автоматизованого проектування з використанням метода динамічної інтеграції даних. Проведено аналіз інтеграції даних і програмних засобів автоматизації на основних етапах процесу проектування. Показано, що автоматизація проектування являє собою набір пов'язаних проектних даних, виражених різними аспектами представлення. Запропоновано нову інформаційну модель середовища автоматизованого проектування, що забезпечує динамічну інтеграцію даних різних аспектів представлення. Запропоновано реляційний метод динамічної інтеграції, що представляє собою взаємодії засобів автоматизації проектування, що дозволяє реалізувати динамічний зв'язок проектних даних і відображення динамічних процесів модифікації методів і умов проектування, який заснований на побудові зв'язків між процедурними складовими проектних даних. На основі розробленого комплексу програмних засобів перевірена на практиці реалізація запропонованого реляційного методу динамічної інтеграції даних в середовищі автоматизованого проектування.

Ключові слова: *динамічна інтеграція, системи автоматизованого проектування, інтегроване середовище*

ABSTRACT

Full amount thesis is 153 pages and contains 24 figures, 10 tables, 7 posters and 23 bibliographic sources for references.

Thesis is devoted to questions of building integrated computer-aided design method using dynamic data integration. Analysis of data integration software tools and automation to the main stages of the design process. It is shown that the automation design is a set of related design data, expressed various aspects of presentation. The new information model of aided design environment provides a dynamic integration of the different aspects of the data presentation. A relational dynamic integration method, which is the interaction of design automation, allowing to implement a dynamic link design data and displaying dynamic modification process, design methods and conditions, which is based on building relationships between procedural elements of design data. Proposed method of relational dynamic data integration has been tested in practice based on the function complex software.

Keywords: *dynamic integration, computer-aided design, integrated environment.*

ЗМІСТ

ВСТУП.....	8
1 САПР ЯК ОБ'ЄКТ ІНТЕГРУВАННЯ	10
1.1 САПР, їх класифікація та структура.....	11
1.2 Порівняння різних САПР.....	29
1.3 Представлення даних в системах автоматизованого проектування.....	41
1.4 Методи інтеграції та інтеграційні властивості САПР	50
1.5 Постановка задачі	63
2 ДИНАМІЧНА ІНТЕГРАЦІЯ ДАНИХ	65
2.1 Інформаційна модель середовища автоматизованого проектування.....	65
2.2 Представлення і взаємодія даних в інформаційній моделі	74
Висновки до розділу	88
3 СТРУКТУРНИЙ СИНТЕЗ ІНТЕГРОВАНИХ САПР.....	89
3.1 Серверний вузол системи та прикладні програми	89
3.2 Опис роботи модуля управління інтегрованої системи	97
3.2.1 Загальна схема роботи	97
3.2.2 Обробка команд користувача.....	101
3.2.3 Робота зі сценаріями	103
3.2.4 Створення нового сценарію проектування.....	104
3.2.5 Редагування сценарію проектування	105
3.2.6 Виконання сценарію проектування.....	106
3.2.7 Робота з тематичними співпроцесорами.....	108
3.2.8 Читання пакету	109
3.2.9 Обробка пакета	110
3.2.10 Регулювання доступу до ЄІП.....	111
3.3 Засоби реалізації динамічної інтеграції даних	112
3.4 Бібліотеки узагальнених операцій і даних. Драйвери.	122
Висновки до розділу	131
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	133
4.1 Характеристика виробничого приміщення.....	133
4.2 Ергономічна безпека	134
4.3 Мікроклімат виробничого приміщення.....	135
4.4 Освітлення	136

4.5	Характеристика випромінювання	139
4.6	Аналіз рівня шуму	139
4.7	Електробезпека	139
4.8	Пожежна безпека	140
4.9	Охорона праці та безпеки в умовах надзвичайних ситуацій	141
	Висновок до розділу	143
	ЗАГАЛЬНІ ВИСНОВКИ	144
	СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	148
	ДОДАТКИ	

ВСТУП

Сучасні інформаційні системи швидко еволюціонують з монолітних сховищ в розподілені додатки, засновані на сервісах з гнучкими схемами використання. Щоб іти в ногу з часом, ІТ організації повинні пристосовувати свої старі додатки до мінливих бізнес-вимог практично в режимі реального часу, і фактично без можливості другого шансу. Сучасний бізнес повинен бути здатний гнучко і швидко пристосовуватися до вимог ринку, але навіть незначні зміни в процесах можуть спричинити переробку безлічі інформаційних систем, які спочатку були розроблені як монолітні сховища. Для збереження конкурентоспроможності, витрати на підтримку повинні постійно знижуватися, а системи постійно еволюціонувати. Описаний в даній роботі підхід робить можливим перехід від монолітних систем до сервіс-орієнтованих систем автоматизованого проектування. Це сприяє гнучкості, слабкій зв'язності, виділенню абстракцій з реальної інфраструктури.

Новизна даного підходу полягає в тому, як моделюється інфраструктура архітектурного рішення, заснована на сервісах, замість фокусування на всьому додатку. Сервіси є маленькими, відокремленими елементами ПЗ, які вирішують одну задачу і можуть бути повторно використані в багатьох додатках. Підхід ґрунтується на принципі слабкої зв'язності, що означає що кожен сервіс - це ізольована сутність з обмеженими зв'язками із іншими загальними ресурсами, такими як бази даних, права доступу до додатків або різні API. Таке архітектурне рішення дозволяє створити рівень абстракції між споживачами і розробниками. Це тягне за собою свободу в реалізації та оновленнях без шкоди для споживачів сервісу. Так, новий підхід має чимало плюсів для бізнесу, але це не найкраще рішення для всіх випадків. Серед плюсів підходу можна виділити наступні пункти:

- природний шлях створення складної модульної системи через інтеграцію сервісів проектування від різних постачальників незалежно від платформи і технології;
- заохочує до слабо зв'язаних систем, допомагає працювати зі старими системами;

- підвищує ефективність шляхом повторного використання сервісів, що знижує витрати і час розробки;
- покращує гнучкість і масштабованість, тому що багато сервісів можуть бути розроблені за допомогою різної компоновки існуючих додатків;
- робить можливим скорочення витрат, пов'язаних з підтримкою рішення;
- можна розробити стандартний протокол спілкування між системами;
- доступ до даних не залежить від географії користувача, можна використовувати мобільні засоби зв'язку;
- можливість поступового введення в експлуатацію частин системи, що дозволяє швидше реагувати на потреби бізнесу.

З мінусів можна виділити:

- важко реалізувати асинхронну взаємодію між сервісами;
- складно і трудомістко реалізувати відповіді в «реальному часі», що впливає на швидкість виконання поставлених завдань;
- різні вразливості в безпеці, пов'язані з процесом обміну даними між багатьма додатками і системами (не актуальне для ізольованих систем);
- складний процес управління транзакціями пов'язаний із логічною взаємодією різних систем.

Впровадження запропонованого підходу для бізнесу не простий процес. Підприємства, що бажають перейти на нові технології повинні бути інформовані про труднощі і супутні проблемах. Цілком очевидно, що перехід буде пов'язаний з багатьма компромісами, і в кожного вони будуть свої.

1 САПР ЯК ОБ'ЄКТ ІНТЕГРУВАННЯ

Система автоматизованого проектування - автоматизована система, що реалізує інформаційну технологію виконання функцій проектування, являє собою організаційно-технічну систему, призначену для автоматизації процесу проектування, що складається з персоналу і комплексу технічних, програмних та інших засобів автоматизації його діяльності. Також для позначення подібних систем широко використовується абревіатура САПР.

Цілі створення і завдання

В рамках життєвого циклу промислових виробів САПР вирішує завдання автоматизації робіт на стадіях проектування та підготовки виробництва.

Основна мета створення САПР - підвищення ефективності роботи інженерів, включаючи:

- скорочення трудомісткості проектування і планування;
- скорочення термінів проектування;
- скорочення собівартості проектування і виготовлення, зменшення витрат на експлуатацію;
- підвищення якості та техніко-економічного рівня результатів проектування;
- скорочення витрат на натурне моделювання та випробування.

Досягнення цих цілей забезпечується шляхом:

- автоматизації оформлення документації;
- інформаційної підтримки та автоматизації процесу прийняття рішень;
- використання технологій паралельного проектування;
- уніфікації проектних рішень і процесів проектування;
- повторного використання проектних рішень, даних і напрацювань;
- стратегічного проектування;
- заміни натурних випробувань та макетування математичним моделюванням;
- підвищення якості управління проектуванням;
- застосування методів варіантного проектування та оптимізації.

1.1 САПР, їх класифікація та структура

Склад і структура

Відповідно до ГОСТ, в структурі САПР виділяють наступні елементи (рис. 1.1):

- **КЗАП САПР** — комплекс засобів автоматизації проектування САПР
 - підсистеми САПР, як елемент структури САПР, виникають при експлуатації користувачами КЗАП підсистем САПР.
 - КЗАП - підсистеми САПР - сукупність ПМК, ПТК та окремих компонентів забезпечення САПР, що не увійшли в програмні комплекси, об'єднана загальною для підсистеми функцією.
 - ПТК — програмно-технічні комплекси
 - компоненти забезпечення ПТК САПР
 - ПМК — програмно-методичні комплекси
 - компоненти забезпечення ПМК САПР
 - компоненти забезпечення САПР, що не увійшли в ПМК и ПТК

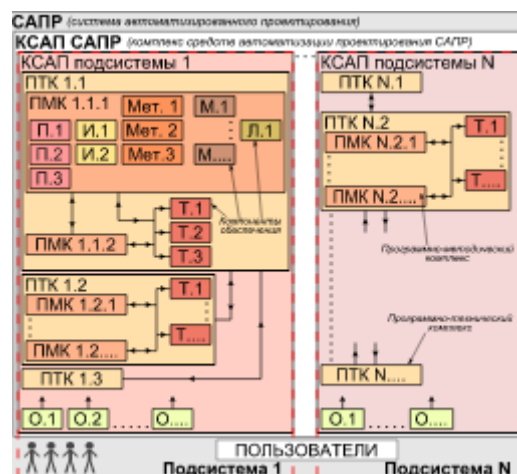


Рис. 1.1. Структура САПР

Сукупність КЗАП різних підсистем формують КЗАП всієї САПР в цілому.

Підсистеми

Згідно до ГОСТ 23501.101-87, складовими структурними частинами САПР є підсистеми, що володіють всіма властивостями систем і створювані як самостійні системи. Кожна підсистема — це виділена за деякими ознаками частина САПР, що забезпечує виконання деяких функціонально-закінчених

послідовностей проектних завдань з отриманням відповідних проектних рішень і проектних документів. За призначенням підсистеми САПР поділяють на два види: проектуючі і обслуговуючі.

- Обслуговуючі підсистеми — об'єктно-незалежні підсистеми, що реалізують функції, загальні для підсистем або САПР в цілому: забезпечують функціонування проектуючих підсистем, оформлення, передачу і виведення даних, супровід програмного забезпечення і т. д., їх сукупність називають системної середовищем (або оболонкою) САПР.
- Проектуючі підсистеми — об'єктно-орієнтовані підсистеми, що реалізують певний етап проектування або групу пов'язаних проектних завдань. Залежно від ставлення до об'єкта проектування, діляться на:
 - об'єктні — виконують проектні процедури та операції, безпосередньо пов'язані з конкретним типом об'єктів проектування.
 - інваріантні — виконують уніфіковані проектні процедури та операції, що мають сенс для багатьох типів об'єктів проектування.

Прикладами проектують підсистем можуть служити підсистеми геометричного тривимірного моделювання механічних об'єктів, схемотехнічного аналізу, трасування з'єднань в друкованих платах [17].

Типовими обслуговуючими підсистемами є:

- підсистеми управління проектними даними
- навчальні підсистеми для освоєння користувачами технологій, реалізованих в САПР
- підсистеми графічного введення-виведення
- система управління базами даних (СУБД).

Компоненти і забезпечення

Кожна підсистема, у свою чергу, складається з компонентів, що забезпечують функціонування підсистеми.

Компонент виконує певну функцію в підсистемі і являє собою найменший (неподільний) самостійно розробляється або покупної елемент САПР (програма, файл моделі транзистора, графічний дисплей, інструкція і т. д.) [20].

Сукупність однотипних компонентів утворює засіб забезпечення САПР. Виділяють наступні види забезпечення САПР[27]:

- технічне забезпечення (ТЗ) - сукупність пов'язаних і взаємодіючих технічних засобів (ЕОМ, периферійні пристрої, мережеве обладнання, лінії зв'язку, вимірювальні засоби);
- математичне забезпечення (МЗ), що об'єднує математичні методи, моделі та алгоритми, використовувані для вирішення завдань автоматизованого проектування. За призначенням і способом реалізації ділять на дві частини:
 - математичні методи і побудовані на них математичні моделі;
 - формалізований опис технології автоматизованого проектування;
- програмне забезпечення (ПЗ). Підрозділяється на загальносистемне і прикладне:
 - прикладне ПЗ реалізує математичне забезпечення для безпосереднього виконання проектних процедур. Включає пакети прикладних програм, призначені для обслуговування певних етапів проектування або рішення груп однотипних завдань усередині різних етапів (модуль проектування трубопроводів, пакет схемотехнічного моделювання, геометричне ядро САПР);
 - загальносистемне ПЗ призначено для управління компонентами технічного забезпечення та забезпечення функціонування прикладних програм. Прикладом компонента загальносистемного ПЗ є операційна система;
- інформаційне забезпечення (ІЗ) - сукупність відомостей, необхідних для виконання проектування. Складається з опису стандартних проектних процедур, типових проектних рішень, комплектуючих виробів і їх моделей, правил і норм проектування. Основна частина ІЗ САПР - бази даних.
- лінгвістичне забезпечення (ЛО) - сукупність мов, що використовуються в САПР для подання інформації про проєктованих об'єктах, процесі і засобах проектування, а також для здійснення діалогу "проектувальник - ЕОМ" і обміну даними між технічними засобами САПР. Включає терміни,

визначення, правила формалізації природної мови, методи стиснення та розгортання;

- в лінгвістичному забезпеченні виділяють клас різного типу мов проектування і моделювання (VHDL, VERILOG, UML, GPSS);
- методичне забезпечення (МетЗ) - опис технології функціонування САПР, методів вибору і застосування користувачами технологічних прийомів для одержання конкретних результатів. Включає теорію процесів, що відбуваються в проєктованих об'єктах, методи аналізу, синтезу систем та їх складових частин, різні методики проектування. Іноді до МетЗ відносять також МЗ і ЛЗ;
- організаційне забезпечення (ОЗ) - сукупність документів, що визначають склад проєктної організації, зв'язок між підрозділами, організаційну структуру об'єкту і системи автоматизації, діяльність в умовах функціонування системи, форму представлення результатів проектування. В ОЗ входять штатні розклади, посадові інструкції, правила експлуатації, накази, положення і т. п.

В САПР, як системі що проєктується, виділяють також ергономічне і правове забезпечення [22].

- Ергономічне забезпечення об'єднує взаємопов'язані вимоги, спрямовані на узгодження психологічних, психофізіологічних, антропометричних характеристик і можливостей людини з технічними характеристиками засобів автоматизації та параметрами робочого середовища на робочому місці.
- Правове забезпечення складається з правових норм, що регламентують правовідносини при функціонуванні САПР, і юридичний статус результатів її функціонування.

Класифікація

ГОСТ 23501.108-85 встановлює наступні ознаки класифікації САПР (рис. 1.2):

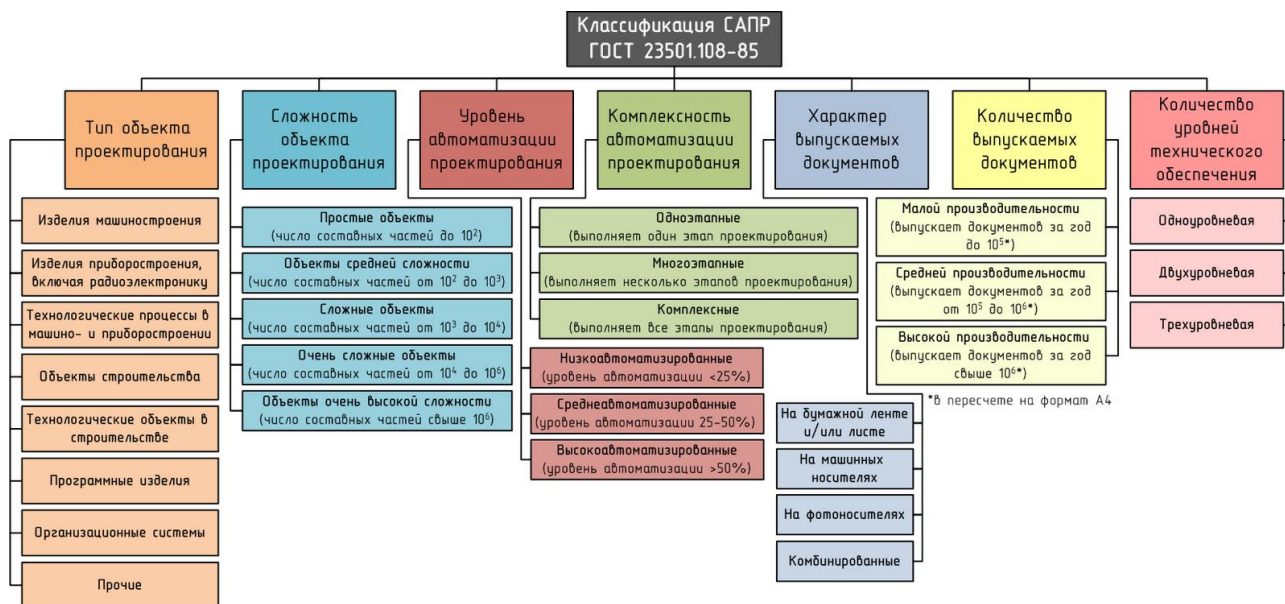


Рис. 1.2. Класифікація САПР

- тип / різновид і складність об'єкта проектування
- рівень і комплексність автоматизації проектування
- характер і кількість випущених документів
- кількість рівнів в структурі технічного забезпечення

Альтернативна класифікація САПР

В області класифікації САПР використовується ряд усталених англomовних термінів, що застосовуються для класифікації програмних додатків і засобів автоматизації САПР за галузевою та цільовим призначенням [31].

За галузевим призначенням

- MCAD (англ. mechanical computer-aided design) — автоматизоване проектування механічних пристроїв. Це машинобудівні САПР, застосовуються в автомобілебудуванні, суднобудуванні, авіакосмічній промисловості, виробництві товарів народного споживання, включають в себе розробку деталей і зборок (механізмів) з використанням параметричного проектування на основі конструктивних елементів, технологій поверхневого та об'ємного моделювання (SolidWorks, Autodesk Inventor, КОМПАС, CATIA);

- EDA (англ. electronic design automation) або ECAD (англ. electronic computer-aided design) — САПР електронних пристроїв, радіоелектронних засобів, інтегральних схем, друкованих плат і т.д. , (Altium Designer, OrCAD);
- AEC CAD (англ. architecture, engineering and construction computer-aided design) або CAAD (англ. computer-aided architectural design) — САПР в галузі архітектури та будівництва. Використовуються для проектування будівель, промислових об'єктів, доріг, мостів та ін. (Autodesk Architectural Desktop, AutoCAD Revit Architecture Suite, Piranesi, ArchiCAD).

За цільовим призначенням

За цільовим призначенням розрізняють САПР або підсистеми САПР, які забезпечують різні аспекти проектування.

- CAD (англ. computer-aided design/drafting) — засоби автоматизованого проектування, у контексті зазначеної класифікації термін позначає засоби САПР, призначені для автоматизації двовимірного та / або тривимірного геометричного проектування, створення конструкторської та / або технологічної документації, і САПР загального призначення.
 - CADD (англ. computer-aided design and drafting) — проектування і створення креслень.
 - CAGD (англ. computer-aided geometric design) — геометричне моделювання.
- CAE (англ. computer-aided engineering) — засоби автоматизації інженерних розрахунків, аналізу та симуляції фізичних процесів, здійснюють динамічне моделювання, перевірку та оптимізацію виробів.
 - CAA (англ. computer-aided analysis) — підклас засобів CAE, використовуваних для комп'ютерного аналізу.
- CAM (англ. computer-aided manufacturing) — засоби технологічної підготовки виробництва виробів, забезпечують автоматизацію програмування і управління обладнання з ЧПУ або ГАПС (Гнучких автоматизованих виробничих систем)..

- CAPP (англ. computer-aided process planning) — засоби автоматизації планування технологічних процесів, які застосовуються на стику систем CAD і CAM.

Багато систем автоматизованого проектування поєднують в собі рішення завдань, що відносяться до різних аспектів проектування CAD / CAM, CAD / CAE, CAD / CAE / CAM. Такі системи називають комплексними, або інтегрованими [21].

За допомогою CAD-засобів створюється геометрична модель виробу, яка використовується в якості вхідних даних в системах CAM і на основі якої в системах CAE формується необхідна для інженерного аналізу модель досліджуваного процесу.

На сьогоднішній день розроблено безліч САПР, які здатні вирішити практично будь-яке завдання проектування в своїй галузі застосування. Нижче наведено список САПР, які сьогодні мають широке застосування при вирішенні завдань автоматизації проектування.

САПР розроблені в країнах СНД

- PSM - САПР із зовнішніх мереж від авторизованого розробника Autodesk компанії IDEAS FOR BUSINESS;
- ADEM — САПР для конструкторсько-технологічної підготовки машинобудівних та металообробних підприємств та програмування обладнання з ЧПУ;
- AutomatiCS — багатокористувацька САПР, призначена для автоматизації проектування, реконструкції та експлуатації систем контролю і управління (КВПіА, АСУТП), обліку енергії, ланцюгів вторинної комутації. Виробництво компанії CSoft Development;
- bCAD — САПР з проектування меблів, торговельного обладнання і дизайну інтер'єрів. Існує також версія для інженерного проектування і безкоштовна студентська версія;
- DipTrace — САПР для проектування друкованих плат. У пакет включено чотири програми: Schematic - розробка принципових схем; PCB Layout -

розводка плат, ручна і автоматичне трасування; ComEdit - редактор корпусів; SchemEdit - редактор компонентів;

- ElectriCS Pro — САПР, призначена для проектування електрообладнання, що застосовується в різних галузях промисловості. Виробництво компанії CSoft Development;
 - EnergyCS — комплекс програм, призначених для виконання електротехнічних розрахунків при проектуванні та експлуатації електроенергетичних систем будь-якої складності. Виробництво компанії CSoft Development;
 - EnergyCS Режим - розрахунок і аналіз встановлених режимів;
 - EnergyCS ТКЗ - розрахунок струмів короткого замикання і струмів замикання на землю;
 - EnergyCS Потери - розрахунок втрат електроенергії в складних енергосистемах;
- GeoniCS — лінійка професійних програмних продуктів, призначених для фахівців у галузі геодезії, геології, землеустрою, проектування генпланів. Виробництво компанії CSoft Development;
 - GeoniCS Дослідження (RGS, RgsPl) призначений для обробки результатів польових вимірювань;
 - GeoniCS Інженерна геологія призначений для обробки даних інженерно-геологічних дослідів;
 - GeoniCS Топоплан-Генплан-Мережі-Траси-Перетини-Геомодель;
 - GeoniCS Plprofile призначений для проектування магістральних нафтогазопроводів;
- IndorCAD — система проектування автомобільних доріг компанії;
- InfrasoftCAD — САПР на основі IntelliCAD від компанії INFRASOFT;
- КЗ — система твердотільного просторового моделювання, розроблена нижегородськими вченими:
 - КЗ-Коттедж — це комплекс комп'ютерних програм для проектування дерев'яних будинків з циліндричних колод і профільованого бруса;

- КЗ-Мебель — це комплекс комп'ютерних програм для виробництва і продажу корпусних меблів. КЗ-Меблі дозволяє автоматизувати процес прийому замовлень та підготовки виробничих завдань;
- КЗ-Тент — комплекс комп'ютерних програм для проектування конструкцій тентів, надає конструктору гранично наочний і найбільш ефективний інструментарій для роботи з поверхнями будь-якої складності. При цьому «КЗ-Тент» забезпечує істотне скорочення термінів по знаходженню форми оболонки, візуалізації кінцевої конструкції, нанесення ліній крою та розгортки шматків на площину;
- КЗ-Ship — це комплекс комп'ютерних програм для виробництва кораблів;
- MechaniCS — додаток до AutoCAD або Autodesk Inventor, призначене для оформлення креслень відповідно до ЕСКД та ін. Виробництво компанії CSoft Development;
- Mineframe — САПР для автоматизованого планування, проектування та супроводу гірничих робіт;
- Model Studio CS — перша російська лінійка програмних продуктів для тривимірного проектування промислових об'єктів. Кожен програмний продукт лінійки дозволяє виконувати компоновану задачу, автоматично виконує розрахунки, генерує специфікації і креслення. Поширюється на платній основі. Виробництво компанії CSoft Development:
 - Model Studio CS ОРУ — програмний продукт призначений для розробки компоновальних рішень в тривимірному просторі відкритих розподільних пристроїв, виконання розрахунків гнучкою ошиновки, випуску проектної та робочої документації (креслень, специфікацій і т. д.);
 - Model Studio CS ЛЭП — програмний продукт призначений для розрахунку і випуску комплекту документів при проектуванні повітряних ліній електропередачі всіх класів напруги на стадіях

будівництва, реконструкції та ремонту. Реалізована унікальна система автоматичного оформлення документів;

- Model Studio CS Молниезащита — програмний продукт призначений для розробки проектів блискавкозахисту в тривимірному просторі, виконання розрахунків зон блискавкозахисту, випуску проектної та робочої документації (креслень, специфікацій і т. д.);
- Model Studio CS Трубопроводи — програмний продукт призначений для розробки компоновальних рішень в тривимірному просторі промислових об'єктів і технологічних установок, випуску проектної та робочої документації (креслень, специфікацій і т. д.);
- Model Studio CS Кабельне господарство - програмний продукт призначений для розробки компоновальних рішень в тривимірному просторі кабельних конструкцій, автоматичної розкладки кабелів і випуску проектної та робочої документації (креслень, специфікацій і т. д.);
- nanoCAD — перша вітчизняна САПР-платформа для вертикальних додатків:
 - nanoCAD free - безкоштовна версія САПР-платформи nanoCAD, що містить інструменти базового проектування і випуску креслень;
 - nanoCAD ВК;
 - nanoCAD Геоника;
 - nanoCAD Конструкції;
 - nanoCAD Механіка;
 - nanoCAD ОПС;
 - nanoCAD СКС;
 - nanoCAD СПДС;
 - nanoCAD Будмайданчик;
 - nanoCAD Схеми;
 - nanoCAD Фундаменти;
 - nanoCAD Електро;
 - nanoCAD Електро ДКС;

- NCTuner — система моделювання твердих тіл для контролю і остаточної настройки керуючих програм для верстатів з ЧПУ;
- NormCAD - програма для розрахунків за будівельними нормами (БНіП) з виведенням звітів з формулами;
- Project Studio CS — лінійка програм для архітектурно-будівельного робочого проектування в середовищі AutoCAD. Виробництво компанії CSoft Development:
 - Project Studio CS СКС;
 - Project Studio CS ОПС;
 - Project Studio CS Електрика;
 - Project Studio CS Водопостачання;
 - Project Studio CS Архітектура;
 - Project Studio CS Конструкції;
 - Project Studio CS Фундаменти;
 - Project Smeta CS — інструмент для складання кошторисів на проектну документацію та вишукувальні роботи в будівництві. Виробництво компанії CSoft Development;
- Raster Arts — растрово-векторна САПР і сучасна векторизація - для сканованих креслень, планів, схем, топографічних і картографічних матеріалів. Виробництво компанії CSoft Development;
- Rubius Electric Suite - лінійка програмних продуктів для автоматизації електротехнічних відділів:
 - Rubius Electric Suite: ЛЕП 0,4-10 кВ - програмний модуль для проектування повітряних ліній електропередачі напругою 0,4 - 10 кВ;
 - Rubius Electric Suite: МЗ - програмний модуль для проектування систем блискавкозахисту;
- SimOne — система моделювання електронних схем. Розробник - компанія Еремекс;
- SprutCAM — професійне рішення для розробки керуючих програм для устаткування з ЧПУ. На сьогоднішній день це єдина російська розробка і

одна з небагатьох серед зарубіжних, підтримуюча у тому числі розробку КП для багатокоординатного токарно-фрезерного обладнання.

- TechnologiCS — спеціалізований програмний продукт, призначений для використання на виробничих підприємствах. Виробництво компанії CSoft Development;
- T-FLEX CAD — САПР (3D и 2D) для машинобудування. Розробник - компанія Топ-Системи. Перша в світі САПР з геометричною параметризацією;
- ТороR — САПР для проектування друкованих плат. Розробник - компанія Еремекс;
- Tropolix — САПР для суднобудування, створена в Петербурзі;
- WinELSO — САПР для проектування систем силового електрообладнання і електроосвітлення, розробка фірми «Російська промислова компанія»;
- БАЗИС (САПР) — комплексна автоматизація проектування технологічної підготовки виробництва і реалізації корпусних меблів;
- ИДЕАЛ-А — безкоштовна програмна оболонка для AutoCAD, призначена для швидкого тривимірного моделювання деталей і автоматизації отримання креслень. Розробник - компанія «Інженерне Дело»;
- КОМПАС — поширена САПР компанії АСКОН у варіантах для двовимірного і тривимірного проектування;
- СПЛИТ — Система проектування лінійного транспорту, програмний комплекс, розроблений компанією «НЕОЛАНТ» і призначений для автоматизації процесу проектування при новому будівництві, реконструкції та капітальному ремонті лінійної частини надземних / підземних, магістральних / промислових нафто- і газопроводів, ліній електропередач;
- САПР «Сударушка» — CAD/CAM/CAE система. Є розвитком системи Гемос (геометричне моделювання обводів літака), розробленої фахівцями російської авіапромисловості в ОКБ ім. А. С. Яковлєва в 1989-1994 р.;
- САПР-ЧПУ — САПР для проектування керуючих програм для верстатів з ЧПУ компанії «Євразія Лімітед»;

- СПДС GraphiCS — додаток до AutoCAD, Autodesk Architectural Desktop, AutoCAD Architecture, призначене для розробки проектно-технічної документації в строгій відповідності з вимогами СПДС. Виробництво компанії CSoft Development;
- САПР НТ "NORMA" — система автоматизованого проектування норм праці (ВАТ «КНІАТ»). Розрахунок режимів різання і норм часу для механообробного виробництва (включаючи верстати з ЧПУ і їх багатOVERстатне обслуговування). Розрахунок норм часу зварювального, гальванічного виробництв. Розрахунок норм часу виготовлення спеціальних пристосувань, штампів холодної та гарячої штамповки, ріжучого і вимірювального інструменту;
- Середовище моделювання МАРС - вітчизняна система автоматизованого моделювання фізично неоднорідних технічних пристроїв і систем МАРС (Моделювання та автоматичний розрахунок систем), заснована на універсальному методі компонентних ланцюгів. Дозволяє моделювати процеси, що протікають в кінематичних, механічних, електричних, електронних, електромеханічних, електроенергетичних, фізико-хімічних системах і пристроях [36];
- СПРУТ — інтегроване метаінструментальне мовне середовище «СПРУТ» (Система Проектування Універсальних Технологій). Середовище для створення наскрізних САПР за принципом RAD-технології (RAPID APPLICATION DEVELOPMENT TOOLS):
 - СПРУТ-ОКП — MES система оперативно-календарного планування, диспетчеризації та контролю виробничого процесу на підприємстві;
 - СПРУТ-ТП — САПР система проектування і нормування технологічних процесів для будь-яких типів виробництв;
- Інформаційно-довідкові системи;
- ІСС "НАВИГАТОР" - інформаційно-довідкова система для Ленінградської АЕС

- Проектна система "Профіль" - проектує зовнішні мережі і будує профілі в AutoCAD;
- NormaCS — інформаційно-довідкова система, містить нормативно-технічну документацію, що діє на території РФ. Розробник - ЗАТ "Нанософт" ;
- Техексперт - інформаційно-довідкова система, містить нормативну-технічну документацію, що діє на території РФ;
- TDMS — система, призначена для управління інформаційними потоками і електронною документацією проектних, конструкторських, виробничих організацій і будь-яких інших підприємств, в роботі яких використовуються технічні дані і створювані на їх основі документи: креслення, плани, схеми, специфікації, відомості і т. п. Виробництво компанії CSoft Development;
- ADEM i-Ris — інформаційно-довідкова система, містить нормативну-технічну документацію для конструкторсько-технологічної підготовки машинобудівних та металообробних виробництв. *САПР іноземних розробників*
Безкоштовні САПР іноземних розробників з відкритим вихідним кодом
- BRL-CAD — відкрита 3D система проектування;
- freeCAD від Aik-Siong Koh (A-S. Koh);
- FreeCAD від Юргена Райгеля (Jürgen Riegel) — відкрита 3D система проектування;
- QCad — відкрита 2D система проектування;
- SALOME — відкрита модульна система 3D проектування;
- Electric — проектування інтегральних схем і електропроводки;
- gEDA — вільна система автоматизованого проектування електроніки;
- KiCad — комплекс для проектування електронних схем та друкованих плат;
- Wings 3D — відкрита програма 3D-моделювання;

Безкоштовні пропріетарні САПР

- Medusa4 — система автоматичного проектування, безкоштовна ліцензія для приватного користування;
- DraftSight — безкоштовна САПР для Windows, Mac і Linux від Dassault Systèmes;

Платные САПР иностранных разработчиков

- 3design CAD — САПР для ювелірного і графічного дизайну;
- Інтермех;
- Cadmech — універсальна САПР в області машинобудування і приладобудування, що розширює функціональні можливості AutoCAD, Inventor, NX, Solid Edge, Pro/ENGINEER, SolidWorks;
- CADElectro — САПР для проектування електрообладнання;
- НПП «Технікон»;
- CADElectro Energy — нова версія САПР на власній графічній платформі, з інтеграцією з ERP, великою базою УДО, автоматизацією типових задач проектування та оформлення конструкторської документації, а також контролем помилок;
- Інтех-Покрій САПР ТП — САПР для автоматизації технології розкрою листового металу;
- Експерт-СКС - САПР для автоматизації на всіх етапах проектування структурованих кабельних систем, ВОЛЗ, ЛВС, лінійних і магістральних мереж;
- Aldec Active-HDL і Riviera — продукти для введення, моделювання та верифікації проектів на мовах VHDL, Verilog, SystemVerilog, SystemC;
- Altium Designer — комплексний пакет розробки електронних систем;
- P-CAD — САПР для проектування електронних пристроїв;
- Allplan — САПР комплексного проектування, проектування всіх розділів в одній системі;
- ANSOFT — САПР електроніки, електромеханіки, систем живлення, управління, зв'язку та радіолокації;
- ArchiCAD — САПР для архітектури компанії Graphisoft;

- Autodesk:
 - AutoCAD — найпоширеніша САПР;
 - Autodesk Inventor — система тривимірного твердотільного проектування для розробки складних машинобудівних виробів;
 - Autodesk Revit — система тривимірного архітектурного і будівельного проектування;
- Aveva:
 - AVEVA Plant — інтегровані рішення для промислових об'єктів, що включає 3D САПР на основі PDMS;
 - AVEVA Marine — інтегровані рішення для суднобудування, що включає 3D САПР;
- Bocad-3D — модульний програмний комплекс для розробки креслень, вузлів і схем металевих і дерев'яних конструкцій в тривимірному просторі. Основним завданням Bocad-3D є деталізація креслень і специфікацій на стадіях КМ і КМД;
- BtoCAD — базова САПР на основі IntelliCAD з форматом DWG і інтерфейсом AutoCAD;
- Cadence;
- Allegro (САПР) — важка САПР для проектування електронних пристроїв;
- OrCAD — САПР для проектування електронних пристроїв;
- Specstra — трасувальник друкованих плат;
- Dassault Systèmes;
- CATIA — САПР для аерокосмічної промисловості;
- SolidWorks — універсальна САПР для машинобудування;
- Dietrichs — німецька САПР / САМ для дерев'яних будівель;
- E3.series — САПР для проектування електричних, гідравлічних і пневматичних схем, схем друкованих плат, креслень джгутів, компоновки шаф. САПР є рішенням в областях: Електротехніки, АСУТП, Енергетики, Транспортного машинобудування (літаки, вертольоти, кораблі, автомобілі), Військово-космічного комплексу (ракети, штучні супутники Землі);

- Electric — проектування інтегральних схем і електропроводки;
- EPLAN — САПР для Електротехніки та АСУТП;
- Foran — спеціалізована суднобудівна система автоматизованого проектування, розроблена фірмою SENER для проектування і будівництва комерційних і військово-морських суден;
- GstarCAD — САПР на основі IntelliCAD, максимально наближена до колишнього інтерфейсу AutoCAD;
- IntelliCAD — DWG-сумісна платформа для САПР. Розробляється міжнародним консорціумом IntelliCAD Technology Consortium. Служить платформою для багатьох САПР, таких як BricsCAD, BtoCAD, CADian, InfracadCAD, GstarCAD, ProgeCAD, ZwCAD та інших;
- IronCAD;
- Ironcad — Професійна система тривимірного твердотілого моделювання та конструювання, а також повнофункціональний 2D CAD. Виділяється серед конкурентів рядом унікальних інструментів. Має вбудований рендер;
- Inovate — Система для тривимірного моделювання та візуалізації. На відміну від IronCAD немає функцій створення креслень і роботи з листовим металом;
- Ironcad Draft — Інструмент для двомірного проектування, зі звичним графічним інтерфейсом користувача і унікальними інтегрованими можливостями роботи з 3D даними;
- Magics — САПР для швидкого прототипування;
- MicroStation — універсальна САПР компанії Bentley Systems (офф. сайт), спочатку створена за технологіями Intergraph Corporation. Основа численних програмних рішень для: ГІС, геодезії, картографії, земельного кадастру, інженерних мереж, проектування електроніки, архітектури, будівництва мостів, автодоріг, будівель і споруд, проектування промислових підприємств і заводів, машинобудування, дизайну інтер'єрів і ін. Основні формати: DGN і DWG. Є безкоштовні версії;

- Morgan MD CAD — інноваційна САПР для конструювання одягу, що включає повний спектр програмних продуктів та інструментів для відцифровки лекал, створення ескізів і проектування моделей, градації лекал, автоматичної розкладки та ін., 3D-моделювання;
- Parametric Technologies Corp. (PTC);
- Pro/Engineer — універсальна САПР для промислових компаній;
- MathCAD — інтегрована система вирішення математичних, інженерно-технічних і наукових завдань;
- CADD5 — потужний машинобудівний пакет САПР;
- PowerShape — САПР англійської компанії Delcam;
- Proteus — САПР проектування електронних пристроїв і друкованих плат;
- Rhinoceros 3D — універсальний САПР для промислового дизайну;
- NX — CAD система для різних галузей промисловості, одна з небагатьох в повній мірі підтримує «пряме» не параметричне моделювання;
- Solid Edge — 2D/3D CAD-система, розроблена Intergraph Corporation в рамках проекту Jupiter, привнесли в ПО новий рівень інтерактивності (інтелектуальний курсор, пряма робота з різними форматами без імпорту / експорту та ін.). Є безкоштовні версії;
- SmartSketch — 2D CAD-система з інноваційним інтерфейсом, розроблена корпорацією Intergraph в рамках проекту Jupiter, привнесли в ПО новий рівень інтерактивності (інтелектуальний курсор, пряма робота з різними форматами без імпорту / експорту та ін.). Деякий час поставлялася Microsoft в пакеті додаткового ПЗ Plus! для Windows. «Молодший брат» системи Solid Edge. Основне призначення - швидкі начерки креслень, створення діаграм, схем, офісні оформлювальні роботи. Містить велику кількість готових компонентів графіки для самих різних областей;
- Tribon — спеціалізована САПР застосовувана в суднобудуванні, розроблена в Швеції;
- TurboCAD — універсальна САПР для архітекторів і конструкторів;
- ZWCAD — одна з САПР на основі IntelliCAD;
- Mastercam - потужна система для токарних і фрезерних 3D робіт;

- Medusa4 — система автоматичного проектування, безкоштовна ліцензія для приватного користування, комерційна ліцензія, ліцензія за комерційне використання - «Платіж за одиничне комерційне використання»;
- JULIVI — Система конструювання та моделювання одягу;
- Tebis - універсальна комплексна САПР для моделювання, обробки і вимірювань.

1.2 Порівняння різних САПР

Оцінка загальних властивостей різних САПР

Всі нові продукти володіють наступними загальними властивостями. По-перше, вони розроблялися з максимальним використанням сучасних операційних систем (Windows, Mac, Linux), що робить їх інтерфейси схожими з іншими програмними продуктами даних операційних систем. Фактично це робиться для автоматизації конструкторської праці, тому користувачі відчують себе комфортно в знайомому середовищі і швидко освоюють новий програмний продукт. Крім того, нові САПР підтримують функції впровадження та зв'язування об'єктів (Object Linking and Embedding - OLE), характерні для офісних пакетів Microsoft. Таким чином, будь-яке зображення тривимірної деталі або пристрою, створене з використанням САПР. Може використовуватися іншими програмами [46].

По-друге, в нових системах використовується компонентний підхід, згідно з яким найважливіші компоненти програмного забезпечення вибираються з доступних програм, після чого розробник системи просто об'єднує перевірені технології, зосереджуючи свою увагу на деталях, що відносяться безпосередньо до проектування. Такий підхід дозволяє скоротити час на розробку, причому користувачеві будуть надані саме ті можливості, які знадобляться йому в процесі проектування конкретного продукту [34].

Таблиця 1.1. Порівняння САПР

Характеристика	CADKEY 97	Euclid Quantum 1.0	Helix Design System V4	I_DEAS Artisan Series 1.0
Ціна	\$1195	\$9995	\$5695	\$4995
Мінімальний обсяг ОЗУ, МБ	16	96	32	64
Розмір після встановлення, МБ	50	2000	150	500
Розмір файлу підкачки, МБ	20	400	120	185
Геометричне ядро	ACIS 2.1	CAS, CADE	DesignBase	Власний
Технологія моделювання	Гібридне моделювання довільних форм	Варіаційне, параметричне, об'єктно-орієнтоване	Варіаційне, параметричне, об'єктно-орієнтоване	Варіаційне, параметричне, об'єктно-орієнтоване
Використання історії	Ні	Часткове	Так	Так
Параметричне моделювання	Ні	Так	Розмірні та геометричні параметри	Так
Двосторонній зв'язок з інтегрованими програмами	Так	Так – з додатками Quantum	Можливість автоматичної розмітки	Так, з інтегрованими додатками
Підтримка конверторів	IGES, DXF, DWG, STL VRML, SAT	IGES, STEP, DXF, VDA	IGES, STL, DXF, DWG, VRML, STEP	DXF, IGES, SET, VDA, STL
Додаткові конвертори	.SAT-конвертори	Користувацькі	Користувацькі	Ні
Підтримка Інтернету	VRML	HTML, VRML	VRML	Ні
Графічний інтерфейс	Windows	Motif, Windows	Власний	Власний
Підтримка OLE 2.0	Тільки контейнер	Так	Так	Ні

Таблиця 1.1. (продовження)

Характеристика	Mechanical Desktop	MicroStation Modeler	Advanced Deigner Package	SolidDesigner
Ціна	\$6250	\$5325	\$26000	\$6500
Мінімальний обсяг ОЗУ, МБ	32	32	64	96
Розмір після встановлення, МБ	62	200	400	200
Розмір файлу підкачки, МБ	64	48	128	96
Геометричне ядро	ACIS 1.6	ACIS 2.1	Власне	Розширене ACIS
Технологія моделювання	Явна, булева, варіаційна, ОО	Явна, булева, параметрична, ОО	Параметрична, ОО	Об'єктно- орієнтоване
Використання історії	Так	Так	Так	Так
Параметричне моделювання	Так	Так	Так	Так
Двосторонній зв'язок з інтегрованими програмами	Так	Так	Так	Ні
Підтримка конверторів	IGES, DXF, DWG, ACIS	IGES, STEP, DWG, ACIS SAT, CADKEY, Versacad	IGES, STL, DXF, DWG, VRML, STEP, TIFF, JPEG, CGM	IGES, SAT, STL
Додаткові конвертори	Сторонні	З використанням ACIS SAT	Catia, CADAM, STEP, IDF	STEP
Підтримка Інтернету	Так	ModelServer Publisher	HTML	HTML
Графічний інтерфейс	Windows	Motif, Windows	Власний	Motif
Підтримка OLE 2.0	Так	Так	Так	Так

Таблиця 1.1. (завершення)

Характеристика	SolidWorks	UniGraphics	Solid Edge
Ціна	\$3995	\$17000	\$5995
Мінімальний обсяг ОЗУ, МБ	48	64	32
Розмір після встановлення, МБ	25	150	75-100
Розмір файлу підкачки, МБ	64	135	150
Геометричне ядро	Parasolid	Parasolid	ACIS 2.0
Технологія моделювання	Параметричне, ОО	Явне, булеве, параметричне, варіаційне, ОО	Параметричне, ОО
Використання історії	Так	Так	Так
Параметричне моделювання	Так	Так	Ні
Двосторонній зв'язок з інтегрованими програмами	Так	Так	Так
Підтримка конверторів	IGES, DXF, DWG, SAT, STL, Parasolid Binary	IGES, STEP, DXF, ACIS SAT, VRML, Parasolid	IGES, STL, DXF, DWG, VRML, STEP, DGN
Додаткові конвертори	Сторонні	CATIA	Сторонні
Підтримка Інтернету	HTML, VRML	VRML	CGM
Графічний інтерфейс	Windows	Motif, ANSI	Windows
Підтримка OLE 2.0	Так	Так	Так

По-третє нові системи засновані на об'єктно-орієнтованій технології, три аспекти якої потрібно розглянути особливо. З точки зору програмування об'єктно-орієнтована технологія означає написання модульних програм таким чином, щоб забезпечити незалежне повторне використання модулів. Типовим об'єктно-орієнтованою мовою програмування є C++. Кожна функція може бути написана на цій мові таким чином, що вона буде діяти як незалежна ціле. Об'єктно-орієнтована технологія визначає також інтерфейс між системою і

користувачем У звичайних системах CAD дані про деталі зазвичай зберігаються в кількох файлах: один файл використовується для геометричної форми, інший - для сітки кінцевих елементів, третій - для траєкторії рухів фрези верстата з ЧПУ, і т.б. В об'єктно-орієнтованих системах всі дані, що відносяться до однієї деталі, зберігаються в одному файлі. При збереженні однакових даних у різних файлах відбувається надлишкове дублювання, а в об'єктно-орієнтованих системах цього вдається уникнути, що призводить до значної економії пам'яті[47].

По-четверте, системи підтримують або параметричне, або варіаційне моделювання. Обидва підходи дозволяють визначати форму, задаючи обмеження, а не характеристики окремих елементів цієї форми. Єдина відмінність в тому, що в одному випадку обмеження враховуються одночасно, а в іншому - послідовно. Прикладом безпосередньої роботи з елементами форми є визначення прямокутника як двох наборів паралельних відрізків, що знаходяться на конкретному відстані один від одного. Однак той же прямокутник може бути визначений за допомогою обмежень, наприклад завданням умови перпендикулярності суміжних відрізків і відстані між паралельними відрізками. Багато системи, що підтримують можливість параметричного або варіаційного моделювання, сприймають очевидні обмеження, такі як перпендикулярність і паралельність, безпосередньо з початкового ескізу користувача, дозволяючи зменшити обсяг даних, що вводяться. У цьому випадку від користувача потрібно тільки ввести розміри, після чого він зможе змінювати форму, варіюючи ці параметри. Така функція системи називається моделюванням за розмірами [46].

Нарешті, в системи вбудовується підтримка спільного проектування через Інтернет. Ця підтримка дозволяє віддаленим користувачам працювати над однією і тією ж деталлю, маючи перед очима її модель на своїх екранах. Розробники також можуть перевіряти проект в цілому, порівнювання свої деталі з деталями інших розробників. Для того щоб скористатися цією можливістю, потрібно як мінімум описати деталь у форматі VRML[53].

Оцінка вимог, що пред'являються САПР до ЕОМ

Для аналізу вимог виберемо 6 найпопулярніших CAD систем на ринку:

- AutoCAD;
- ArchiCAD;
- КОМПАС;
- SolidWorks;
- T-FLEX CAD;
- Nano CAD;

AutoCAD (сайт виробника - www.autodesk.ru) - двох- і тривимірна система автоматизованого проектування і креслення, розроблена компанією «Autodesk» (США) [45].

Функціональні можливості: використання елементарних графічних примітивів в області двовимірного проектування для отримання складних об'єктів; надання широких можливостей роботи з шарами і анотативними об'єктами; використання механізму зовнішніх посилань дозволяє розбивати креслення на складові файли, а також використання динамічних блоків розширюють можливості автоматизації 2D-проектування звичайним користувачем без використання програмування; підтримка двовимірного параметричного креслення; можливість динамічного зв'язку креслення з реальними картографічними даними; управління тривимірної печаткою та підтримка хмар точок і багато іншого [57].

Останньою на даний момент версією є AutoCAD 2014, яка вийшла в 2013-му році. Для ефективної роботи з програмою AutoCAD 2014 необхідно враховувати вимоги, представлені у таблиці:

Таблиця 1.2. Системні вимоги для AutoCAD 2014

Операційна система	Microsoft Windows 8 (64-біт и 32- біт), Microsoft Windows 7 (64- біт і 32- біт), Microsoft Windows XP (64- біт і 32- біт)
Процесор	Для 32-разрядной версії необхідно: • для <i>Windows 7</i>: процесор Intel Pentium 4 або двоядерний процесор AMD Athlon з тактовою частотою 3 ГГц або вище, з підтримкою SSE2; • для <i>Windows XP</i>: процесор Pentium 4 або двоядерний процесор AMD Athlon з тактовою частотою 1,6 ГГц або вище, з підтримкою SSE2. Для 64-розрядної версії потрібно процесор AMD Athlon 64, AMD Opteron, Intel Xeon з підтримкою Intel EM64T або Intel Pentium 4 з підтримкою Intel EM64T (усі - з підтримкою SSE2).
Пам'ять	Потрібно 2 ГБ оперативної пам'яті (рекомендується 4 ГБ).
Простір на жорсткому диску	6 ГБ для повного встановлення.
Монітор	Мінімальне розширення не менше 1024 x 768 і підтримка режиму true color (рекомендовано 1600 x 1050 і вище).

ArchiCAD (сайт виробника - www.archicad.ru) — програмне забезпечення для архітектурно-будівельного проектування розробленого компанією «Graphisoft» (США).

Функціональні можливості: розробка повного набору документації по проекту в одному файлі; використання концепції віртуального будівлі; можливість вилучення різноманітної інформації про спроектованому об'єкті: поверхові плани, фасади, розрізи, експлікації, специфікації, презентаційні матеріали; підтримка взаємодії з різними інженерними програмами через формат IFC, унікальні інструменти для організації спільної роботи в рамках однієї проектною групи (вони дуже ефективні при роботі з масштабними проектами, коли, працюючи з великим комплексом завдань, проектантів істотно обмежує необхідність узгоджувати роботу, як архітекторів, так та інженерів) і багато іншого [48].

Останньою на даний момент версією є ArchiCAD 17, яка вийшла в 2013-му році. Для ефективної роботи з програмою ArchiCAD 17 необхідно враховувати вимоги, представлені в таблиці.

Таблиця 1.3. Системні вимоги для ArchiCAD 17

Операційна система	Windows 8 (64- біт) Windows 7 (64- біт) Windows Vista (64- біт)	Macintosh® OS X 10.8 Mountain Lion Macintosh® OS X 10.7 Lion Macintosh® OS X 10.6 Snow Leopard
Процесор	Потрібно 64-розрядний процесор Intel Core2 Duo або вище.	
Пам'ять	Потрібно 4 ГБ оперативної пам'яті. Рекомендується використовувати 8 ГБ і більше для складних моделей.	
Простір на жорсткому диску	Для повної установки програми потрібно мінімум 5 Гб. Додатково для роботи зі складними моделями і тривимірної візуалізації потрібно близько 10 Гб.	
Монітор	Мінімальне розширення: 1024x768. Рекомендоване розширення: 1280x1024 або вище.	
Відеокарта	Підтримка режиму OpenGL 2.0 з об'ємом відеопам'яті не менше 256 МБ. Рекомендується використання відеокарт з об'ємом відеопам'яті 512 МБ і більше	

SolidWorks (сайт виробника - www.solidworks.ru) — система автоматизованого проектування, інженерного аналізу та підготовки виробництва виробів будь-якої складності і призначення, продукт розроблений компанією «SolidWorks Corporation» (США).

Функціональні можливості: використання гібридного параметричного моделювання; створення мультиплікації (анімації) на основі 3D моделей; оформлення креслень по ЕСКД; проектування зборок різної складності; наявність експертних систем; проектування виробів з урахуванням специфіки виготовлення; робота з імпортованою геометрією; редагування на основі параметрів та історії побудови моделі; "пряме" редагування та багато іншого[32].

Останньою на даний момент версією є SolidWorks 2014, яка вийшла в 2013-му році. Для ефективної роботи з програмою SolidWorks 2014 необхідно враховувати вимоги, представлені у таблиці.

Таблиця 1.4. Системні вимоги для SolidWorks 2014

Операційна система	Windows Vista, 7, 8 (Windows 8 тільки 64-розрядна).
Процесор	Intel і AMD процесори з підтримкою технології SSE2 (рекомендуються ОС 64bit).
Пам'ять	Потрібно мінімум 2Гб RAM або більше.
Простір на жорсткому диску	Для повної установки програми потрібно 5 Гб або більше.
Монітор	Мінімальне розширення: 1024x768. Рекомендоване розширення: 1280x1024 або вище.
Відеокарта	Рекомендуються сертифіковані графічні карти для робочих станцій з підтримкою OpenGL і протестовані драйвери до них.

КОМПАС (сайт виробника - www.kompas.ru) — система автоматизованого проектування, розроблена російською компанією «АСКОН» (Росія).

Функціональні можливості: автоматична генерація асоціативних видів тривимірних моделей (асоціація з моделлю: зміни в моделі призводять до зміни зображення на кресленні); синхронізація даних в основному написі креслення з даними з тривимірною моделі; можливість зв'язку тривимірних моделей і креслень зі специфікаціями, тобто при «належному» проектуванні специфікація може бути отримана автоматично; зміни в кресленні або моделі можуть передаватися в специфікацію, і навпаки; наявність великої кількості додаткових бібліотек до програм сімейства, що автоматизують різні спеціалізовані завдання і багато іншого [63].

Останньою на даний момент версією є КОМПАС 3D V14, яка вийшла в 2013-му році. Для ефективної роботи з програмою КОМПАС 3D V14 необхідно враховувати вимоги, представлені у таблиці.

Таблиця 1.5. Системні вимоги для КОМПАС 3D V14

Операційна система	Windows 8, 7 SP1 і вище, Vista SP2 і вище, XP SP3 (32-розрядна).
Процесор	Pentium III з тактовою частотою 800 МГц.
Пам'ять	Потрібно 512 Мб оперативної пам'яті.
Простір на жорсткому диску	Потрібно мінімум 3 Гб.
Монітор	Мінімальне розширення: 1024x768. Рекомендоване розширення: 1280x1024 або вище.

Система T-FLEX CAD (сайт виробника - www.tflex.ru) - професійна конструкторська програма, розроблена компанією «Топ системи» (Росія).

Функціональні можливості: використання широких засобів для автоматизації проектування, спеціальних інструментів для роботи з великими збірками, єдиної документної структури; можливість вести колективну розробку; оформлення і управління непараметричних і складальними кресленнями; наявність системи підготовки специфікацій і багато іншого [70].

Останньою на даний момент версією є T-FLEX CAD 12, яка вийшла в 2012-му році. Для ефективної роботи з програмою T-FLEX CAD 12 необхідно враховувати вимоги, представлені у таблиці [48].

Таблиця 1.6. Системні вимоги для T-FLEX CAD 12

Операційна система	Windows XP, Windows 7 (64-біт), Windows 8 (64- біт)
Процесор	Core i7 або аналог
Пам'ять	Мінімум 2 Гб. Рекомендований обсяг 8 Гб і більше.
Простір на жорсткому диску	Для повної установки програми потрібно 5 Гб або більше.
Монітор	Мінімальне розширення: 1024x768. Рекомендоване розширення: 1280x1024 або вище.
Операційна система	NVIDIA або AMD з пам'яттю 1Гб і вище і підтримкою OpenGL 2.0 і вище

nanoCAD (сайт виробника - www.nanocad.ru) – перша вільно розповсюджується базова САПР-платформа для різних галузей, розроблена компанією «Нанософт» (Росія).

Функціональні можливості: пряма підтримка DWG (креслення, розроблені в *nanoCAD*, можна відкрити в середовищі AutoCAD без додаткових перетворень, і навпаки, креслення, розроблені в середовищі AutoCAD, відкриваються в середовищі *nanoCAD*); включені інструменти, які дозволяють працювати з растровими підкладками (редагувати не тільки загальні параметри растра, а й усувати перекоси, вирівнювати зображення, прати частину зображення ластиком, або навпаки допрацьовувати зображення за допомогою інструменту Олівець); володіє унікальною можливістю прив'язуватися до даних на монохромному растровому зображенні і багато іншого [47].

Останньою на даний момент версією є *nanoCAD free 5.1*, яка вийшла в 2013-му році. Для ефективної роботи з програмою *nanoCAD free 5.1* необхідно враховувати вимоги, представлені у таблиці 1.7.

Таблиця 1.7. Системні вимоги для *nanoCAD free 5.1*

Операційна система	Microsoft® Windows8; Microsoft® Windows 7; Microsoft Windows Vista; Microsoft Windows XP.
Процесор	Intel Pentium 4 або аналогічний AMD Athlon або вище.
Пам'ять	Від 512 Мб, рекомендовано 2 Гб і більше при роботі з великими проектами
Простір на жорсткому диску	Для повної установки програми необхідно близько 400 Мб. Додатково для роботи необхідно 1-3 Гб.
Монітор	Мінімальне розширення: 1024x768. Рекомендоване розширення: 1280x1024 або вище.
Операційна система	Відеоадаптер з OpenGL-сумісної апаратної 3D акселерації.

Таким чином, порівняльний аналіз систем проектування за основними характеристиками представлений в таблиці 1.8.

Таблиця 1.8. Порівняльна характеристика САПР

САПР	Порівняльні характеристики				
	Фірма-виробник	Розповсюдження	Простота інтерфейс	Мова інтерфейсу	Остання версія (рік виходу)
AutoCAD	Autodesk	Ціна: 1680\$. Наявність безкоштовної версії для навчання	+	Англійська Російська	AutoCAD 2014 (2013 г.)
КОМПАС	Аскон	Ціна: 2200\$.	+	Тільки російська	КОМПАС 3D V14 (2013 г.)
SolidWorks	SolidWorks Corporation	Ціна: 4649 \$. Наявність безкоштовної версії для навчання	-	Багатомовний	SolidWorks 2014 (2013 г.)
ArchiCAD	Graphisoft	Ціна: 1000 \$.	+	Англійська Російська	ArchiCAD 17 (2013 г.)
T-FLEX CAD	Топ Системы	Ціна: 1900\$.	+	Тільки російська	T-FLEX CAD 12 (2012 г.)
nanoCAD	Нанософт	Безкоштовно	-	Тільки російська	nanoCAD free 5.1 (2013 г.)

1.3 Представлення даних в системах автоматизованого проектування

Дані в процесі проектування відіграють важливу роль. Проектувальник, створюючи проект, формує пов'язані набори різних даних, створюючи тим самим цілісний опис об'єкта проектування. В даний час в різних галузях проектування широке застосування отримали САПР, що полегшують і автоматизують працю проектувальника.

САПР використовуються для вирішення завдання проектування об'єктів зі структурованим представленням даних. Проектування об'єктів зі структурованим представленням даних є процесом формування пов'язаного опису об'єкта з набору стандартних і унікальних елементів. Застосування елементів накладає на процес проектування певні умови, пов'язані з визначенням зв'язків між використовуваними при проектуванні компонентами [47].

Очевидно, що процес проектування будь-якого об'єкта не зводиться до обробки виключно структурованих даних. Створення зв'язкового опису об'єкта на основі первісного загального і досить невизначеного завдання на проектування і є, власне, процесом проектування. При цьому ступінь структурованості даних зростає в ході реалізації проекту. Застосування САПР стає можливим при певній мірі структурованості проектних даних [33].

Можливі наступні критерії, відповідно до яких задача проектування може бути віднесена до класу задач з структурованим представленням даних:

- проектування об'єкту проводиться в основному зі стандартних деталей і складових;
- застосування нестандартних деталей зведено до мінімуму,
- нестандартні деталі і складові проектуються сторонніми організаціями чи групами проектувальників не пов'язаних з проектуванням основного об'єкта;
- нестандартні деталі, використовувані при проектуванні, приймаються як єдині неподільні компоненти з певними і не змінюються в ході проектування робочими параметрами.

САПР рясно застосовується на п'яти основних етапах проектування, а саме: отримання проектного завдання, безпосереднє проектування об'єкта, підготовка

технічної документації, контроль монтажу та введення в експлуатацію і безпосередньо експлуатація об'єкта. В даному випадку етапи проектування становлять інтерес з точки зору використання та обробки даних на кожному етапі проектування [54].

Розглянемо як приклад завдань з структурованим представленням даних завдання проектування будь-якого промислового об'єкта. Процес проектування об'єкта зазвичай розбивається на етапи, на яких проводиться обробка різних видів даних:

1. Проектне завдання:

- 1.1. Графічна інформація: креслення, схеми;
- 1.2. таблиці даних;
- 1.3. Текстовий опис завдання.

2. Проектування об'єкта:

- 2.1. Проектування креслень і геометричне моделювання;
- 2.2. Проведення інженерних розрахунків;
- 2.3. Технологічна підготовка виробництва;
- 2.4. Рішення задач оптимізації;
- 2.5. Побудова імітаційного стенду для перевірки параметрів об'єкта.

3. Підготовка технічної документації:

- 3.1. Проектні креслення і схеми;
- 3.2. Таблиці результатів проектування та специфікації;
- 3.3. Пояснювальна записка до проекту.

На першому етапі проектування виробляється формування технічного завдання. При формуванні технічного завдання проводиться збір первинної інформації про об'єкт для проектування. На цьому етапі проводиться первинне структурування даних по відповідним типам. Креслення та схеми проектованого об'єкта являють собою графічну інформацію, тобто ту інформацію, яка обробляється за допомогою засобів обробки графіки [59].

Списки устаткування що при проектуванні об'єкта являють собою табличну інформацію, тобто інформацію, оброблювану різними засобами управління баз даних (СУБД). Отже, формуються таблиці встановленого

обладнання з необхідними даними для проектування об'єкта проектування. В якості допоміжних даних для проектування додаються значення деяких параметрів, використовуваних проектувальниками надалі при виконанні необхідних математичних обчислень. Ці дані відносяться до текстового типу і обробляються відповідно засобами обробки текстової інформації (текстовими процесорами) [64].

Другим етапом проектування об'єкта є безпосереднє проектування схеми з'єднання компонентів, складання схеми електропостачання, вибір комутаційної і захисної апаратури, а також системи охолодження. На даному етапі проектування проводиться безпосереднє рішення задачі проектування. Розглянемо відповідність даного завдання критеріям визначення класу задач з структурованим поданням даних.

Першим критерієм відбору завдань проектування відповідають ситуації проектування, коли використовуються тільки стандартні типи конструктивних елементів, захисна апаратура та трансформатори вибираються тільки зі стандартних апаратів, вироблених на різних підприємствах. Проектування і виробництво спеціальної електричної апаратури електромонтажною проектною організацією не виконуються. Згідно з другим критерієм відбору проектні організації використовують нестандартні компоненти рідко і лише в крайніх випадках. Відповідно до третього критерієм відбору електромонтажні проектні організації замовляють нестандартні компоненти світильників у сторонніх спеціалізованих організаціях або створюють власними силами, але процес проектування таких компонентів не включається в процес проектування об'єкта, а виконується окремо. Відповідно до четвертим критерієм відбору готові нестандартні компоненти включаються в проект нарівні зі стандартним обладнанням. З наведеного вище відповідності всім критеріям випливає висновок, що завдання проектування об'єкта з певного рівня опрацювання відноситься до класу задач з структурованим представленням даних [66].

Третім етапом проектування є підготовка технічної документації проекту. На цьому етапі проводиться формування звітів про результати проектування в електронному або паперовому вигляді.

На кожному етапі проектування об'єкта обробляються креслення схем електропостачання, креслення друкованої плати, таблиці вихідних значень, довідкові таблиці, текстові описи проекту та набори математичних виразів і формул, що використовуються при виконанні розрахунків в процесі проектування [34].

При використанні САПР в процесі проектування проектна документація інтерпретується у вигляді чотирьох аспектів представлення даних:

- Графічний - графічне зображення елементів системи, що використовується для формування графічної документації;
- Табличний - таблиці і специфікації елементів системи;
- Текстовий - текстовий опис елементів системи і пояснювальні записки до проекту;
- Математичний - набір формул і виразів, що використовуються для вибору і визначення елементів системи.

Графічно аспект представлення даних виражається у формуванні креслень і схем проекту оброблюваних графічними засобами автоматизації проектування.

Табличний аспект представлення даних виражається у формуванні таблиць, специфікацій і зведених відомостей устаткування оброблюваний СУБД.

Текстовий аспект представлення даних виражається у формуванні текстового опису проекту оброблюваний різними текстовими процесорами [52].

Математичний аспект представлення даних виражається у формуванні та застосуванні різних формул і математичних виразів в процесі проектування.

Кожен аспект подання є результатом вирішення частини поставленого завдання, і в сукупності вони являють собою відображення повної картини вирішення завдання проектування [48].

Всі складові компоненти системи визначені, і система є логічно завершеною. Отже, вони мають набір вхідних і вихідних параметрів, а також механізм перетворення вхідних параметрів у вихідні. Процес проектування об'єктів даного типу являє собою зв'язування між собою всіх компонентів системи з певною функціональною залежністю. Звідси випливає: всі компоненти

системи пов'язані причинно-наслідковими відносинами, що приводять до вирішення завдання.

Процес проектування являє собою завдання з відомою кінцевою метою. Причому в процесі проектування проектувальник здійснює рух до кінцевої мети проектування послідовно крок за кроком. На кожному етапі проектування проектувальник отримує дані необхідні для виконання конкретного кроку. Можливість побудови "рамкової конструкції", у розумінні її як послідовності подій, в понятті процесу проектування відсутня. Тобто, описавши початкове і кінцеве дію, початкові та кінцеві дані, а також вхідні і вихідні параметри отримати повноцінний процес проектування не можливо. Звідси випливає: всі дії процесу проектування виробляються послідовно від початкового етапу до кінцевого і призводять до кінцевої мети проектування [62].

Виконуючи процес проектування, проектувальник будує логічну послідовну ланцюжок дій. Кожна дія задовольняє першим двом принципам. Але в процесі проектування, якого-небудь об'єкта іноді виникає необхідність внести деякі зміни і доповнення в існуючі рішення. Причому дуже часто зміни кардинально змінюють структуру рішення. У такому випадку проектувальнику необхідно переконатися в тому, що будь-яка модифікація структури рішення не руйнує причинно-наслідкові зв'язки компонентів проекту і призводить до тієї ж самої кінцевої мети проектування. Тобто не порушується перший і другий принцип повноти вирішення даного класу задач. Звідси випливає: будь-які модифікації структури рішення задачі не повинні порушувати зв'язок всіх компонентів системи [43].

Елемент проекту в процесі проектування має двоїсте опис. Декларативні дані P відображають статичний стан об'єкта, описуючи параметри і характеристики. Процедурні дані T описують можливі дії над об'єктом. Процедурна складова даних об'єкта представляє в САПР та інших засобах автоматизації власне процес створення проектної документації. T реалізується безліччю команд, використовуваних програмних процесорів.

В САПР різних типів використовуються різні формати запису даних. Наприклад, графічні дані в різних САПР записуються як у вигляді векторів

(AutoCAD, Компас), так і у вигляді об'єктів (SolidWorks). Ці формати запису відображають декларативну складову проектних даних P . Таблиці, текст, математичні формули по визначенню описують об'єкт і тим самим входять в P .

$$P = P_g * P_{tb} * P_b * P_m, \quad (1.1)$$

де P_g — графічний, P_{tb} — табличний, P_b — текстовий, P_m — математичний аспекти представлення даних.

Кожен з аспектів представлення даних містить в собі сукупність даних усіх типів. Графічний формат подання поряд з векторами містить дані цілочисельного і дійсного типу. У табличному форматі представлення поєднуються всі типи даних, починаючи з цілочисельного і символьного і закінчуючи динамічними показниками на область пам'яті комп'ютера. У текстовому поданні відображаються в основному типи даних, які мають лінгвістичне уявлення, наприклад, цілочисельний або символьний тип. Хоча останнім часом багато текстові процесори мають можливість обробляти і багато інших типів, але про це трохи пізніше [62].

Кожен з аспектів представлення даних містить в собі сукупність даних усіх типів. Графічний формат подання поряд з векторами містить дані цілочисельного і дійсного типу. У табличному форматі представлення поєднуються всі типи даних, починаючи з цілочисельного і символьного і закінчуючи динамічними показниками на область пам'яті комп'ютера. У текстовому поданні відображаються в основному типи даних, які мають лінгвістичне уявлення, наприклад, цілочисельний або символьний тип. Хоча останнім часом багато текстові процесори мають можливість обробляти і багато інших типів, але про це трохи пізніше [44].

Оскільки компоненти проекту зв'язуються деякої розрахунково-аналітичної операцією, потрібно її опис на етапі проектування об'єкта. Розрахунково-аналітична операція являє собою відношення між собою вхідних і вихідних даних об'єкта. В основному, в даному класі задач таким ставленням є функції порівняння та математичні формули, що вживаються при проектуванні.

Найкращим способом опису розрахунково-аналітичних відносин такого роду є математичний опис [26].

Математичний опис являє собою набір формул, поєднання яких представляє собою механізм скріплення об'єктів проекту через їх вхідні і вихідні дані. Причому вихідні дані об'єкта є результатом розрахунково аналітичної операції над вхідними даними. При зв'язуванні об'єктів між собою за допомогою розрахунково-аналітичної операції вхідні дані одного об'єкта є вихідними даними іншого.

Математичний опис проекту не має свого власного формату в комп'ютерному поданні. Будь-яке математичне вираз можна записати у текстовому вигляді. Розрахунково-аналітичні операції пов'язують не узагальнене уявлення даних як графічне, табличне, текстове, а подання даних певних типів оброблюваних комп'ютером, такі як, наприклад, цілочисельний тип, дійсний тип, символний тип і так далі [26].

Вирішуючи завдання безпосереднього фізичного зв'язування об'єктів проекту розрахунково-аналітичними операціями, слід в першу чергу орієнтуватися не на узагальнене подання даних, а на вхідні в нього типи. Розрахунково - аналітичні операції використовують в основному цифрові типи такі, як речовинний і цілочисельний, і символні типи. Зазначені типи даних використовуються в усіх представлених даних і методи роботи з ними будуть описані трохи нижче.

Багато формати представлення даних використовуються різними програмними засобами різних виробників, і тому ці формати стали свого роду стандартними. Такі як, наприклад, dwg - формат подання графічного зображення системою AutoCAD фірми Autodesk і Компас, db і dbf формати представлення табличних даних системами dBase, Clipper, Paradox, sql формат представлення реляційних баз даних MySQL, MSSQL, doc та rtf формати представлення текстових даних в текстових процесорах Word фірми Microsoft і багато інших.

У проектуванні застосовуються засоби обробки графічних даних різних типів. Формати подання графічних даних також діляться на кілька типів [37].

Перший тип являє собою формати растрового зображення. Растрове зображення складається з набору пікселів (точок) певної величини мають певні характеристики. Кожен піксель в залежності від формату має різну величину, починаючи від розміру точки екрана монітора залежно від його дозволу.

Другий тип являє собою формати векторного зображення. Векторне зображення складається з ліній мають певні параметри. Такими параметрами є граничні координати лінії, вид лінії, а також у деяких форматах використовується і ознака напрямку лінії. Векторний тип призначений в основному для виконання будь-яких графічних побудов і твори певних операцій над об'єктами [24].

Табличне представлення даних за способом використання поділяється на два основних типи: реляційне і не реляційне. Чи не реляційна база даних являє собою одну таблицю, розміщену в одному файлі, і всі зв'язки і посилання між полями таблиці розміщуються в цьому ж файлі. Реляційна база даних являє собою набір декількох таблиць розміщуються кожна у своєму файлі і пов'язаних між собою різними видами зв'язків (наприклад, один до багатьох, безліч до одного). Причому всі зв'язки в реляційній базі даних записуються в окремі файли, кількість яких регламентується ієрархією бази даних і кількістю файлів таблиць [23].

Ефективність використання того чи іншого типу баз даних визначається індивідуально для кожного завдання. Використання не реляційної бази даних вигідно в тому випадку, коли структура створеної таблиці ніколи не буде змінюватися, і всі дії з базою даних такого типу будуть носити примітивний характер. У такому випадку один компактний файл бази даних краще. Останнім часом, незважаючи на зазначені перевага не реляційної бази даних для ряду завдань, у світі проявилася тенденція до формування баз даних тільки реляційного типу. Дійсно в постійно прискорюється темпі роботи виникає необхідність вирішувати більш складні завдання, чому успішно сприяє архітектура реляційних баз даних.

Процедурне подання даних відображає сутність об'єкта в описі взаємозв'язку декларативного представлення даних компонентів проектного

об'єкта. Опис процедурного представлення даних всіх компонентів об'єкта являє собою набір функцій і математичних залежностей, що пов'язують компоненти в єдиний об'єкт. Отже, рішенням задачі проектувальника побудови об'єкта з компонентів є зв'язування процедурного представлення даних з декларативним поданням даних об'єкта [17].

У розглянутих САПР використовуються команди побудови стандартних елементів з безлічі X , причому зазвичай формати запису аналогічних команд в різних САПР збігаються. Записи команд обробки графіки відображають процедурну складову проектних даних T . Команди обробки таблиць, тексту, математичних обчислень також входять в T .

$$T = T_g * T_{tb} * T_b * T_m, \quad (1.2)$$

де T_g — множина команд графічного процесора, T_{tb} — множина команд табличного процесора, T_b — множина команд текстового процесора, і T_m — множина команд математичного процесора.

Рішенням задачі проектування є побудова пов'язаної системи описів елементів схеми. Оператор K реалізує зв'язок між описами об'єктів з набору стандартних елементів X . Наприклад, стандартним елементам «точка» і «лінія» може відповідати оператор «починається», який реалізує вираз «лінія починається з точки» [19].

В узагальненому вигляді система описів об'єкта проектування виглядає наступним чином:

$$G = (P, T, K), \quad (1.3)$$

де P , T — множини описів об'єкта проектування, і K — множина однотипних відносин між описами об'єкта.

Наявність в G складової T створює передумови для об'єднання (інтеграції) процедурних даних створюваного об'єкта, оброблюваних зараз у різних системах і засобах автоматизації.

Таким чином, наприклад, лінія, будучи одним з елементів монтажного креслення, у викладеній термінології відбивається графічним аспектом P_g

декларативних складових P проектних даних з набору X стандартних елементів. Команда програмного процесора, що реалізує побудову лінії відбивається графічним аспектом Tg процедурних складових T проектних даних з набору X стандартних елементів, з яких і формується опис проєктованого об'єкта. Побудова лінії щодо інших стандартних елементів креслення регламентується безліччю операторів зв'язку P , що забезпечують взаємодію елементів креслення з набору X стандартних елементів. Аналогічно і для інших аспектів представлення даних.

Отже, в сучасних САПР для завдання даних реалізовані різні способи, адекватні проектної документації, і створені передумови для розвитку методів і засобів динамічної інтеграції даних.

1.4 Методи інтеграції та інтеграційні властивості САПР

Щоб більш розважливо підійти до проблеми вибору інтеграційного рішення, треба чітко визначити свої бізнес-завдання, а також розуміти загальні принципи і підходи інтеграції. В цілому підходи до інтеграції діляться на три основних види інтеграції: інформаційно-орієнтована, сервісно-орієнтована і процесно-орієнтована, так само як гібриди цих підходів.

Інформаційно-орієнтована інтеграція

Інформаційно-орієнтована інтеграція використовується в тих випадках, коли необхідно тільки реплікувати інформацію між двома або більше системами. Інформаційно - орієнтована інтеграція менш дорога і складна, ніж інші види інтеграції, тому що інформація просто витягується з вихідної системи, перетворюється, щоб зняти семантичні відмінності, і передається в потрібну систему. Технологія інформаційно-орієнтованої інтеграції включає брокери повідомлень (SeeBeyond і WebMethods), BO middleware (IBM MQSeries), сервери реплікації баз даних та інші технології, які мають справу з поширенням інформації між двома або більше системами. Багато в чому завдяки успіху продуктів, створених на основі реляційних баз даних і супутніх стандартів (таких, як SQL і ODBC), інтеграція на рівні даних продовжує панувати в якості способу оптимізації взаємозв'язків між різними системами [48].

Сервісно - орієнтована інтеграція

Сервісно-орієнтована інтеграція потрібна для тих завдань, які повинні спільно використовувати і функції додатків, та інформацію. Цей підхід вже дозволяє додаткам отримувати доступ до функцій інших додатків. Ідея полягає в тому, щоб використовувати прикладні сервіси, які вже існують, а не створювати їх щоразу заново. Замість спеціалізованих інтерфейсів між окремими додатками ПЗ, що реалізує цей підхід, покладається на сполучну середу з можливістю багаторазового використання, яка грає роль універсального програмного ядра, що з'єднує всі програми [50].

Процесно-орієнтована інтеграція

Процесно-орієнтована інтеграція надає можливість приєднатися до внутрішніх прикладних процесів кожного додатка, причому таким чином, щоб не просто використовувати його функції, а створити новий або мета-процес, який і зв'яже програми. Наприклад, у компанії є система, яка автоматизує процес, пов'язаний зі створенням виробу, система, яка автоматизує продаж виробу, і система підтримки логістики та доставки виробу замовнику. Процесно-орієнтована інтеграція повинна зв'язати ці процеси, автоматизуючи роботу системи в цілому, при цьому створюючи головний процес, який охоплює багато систем. Різниця між цим підходом та інформаційно-орієнтованою інтеграцією в тому, що інтеграція відбувається саме шляхом створення нового спільного бізнес-процесу, тоді як інформаційно-орієнтований підхід підключає програми через ряд інформаційних потоків. По суті, це перенесення логіки інтеграції ще на один рівень абстракції вгору. Технологія процесно-орієнтованої інтеграції забезпечує інший рівень абстракції (інший спосіб представити інформаційні потоки або запити про надання сервісів), керуючи тим, як системи спільно використовують і інформацію, і сервіси [48].

Методики створення складних інтегрованих систем

Інтеграція "кожен з кожним"

Це традиційний підхід до інтеграції систем, що полягає у створенні спеціалізованих інтерфейсів обміну даними для кожної пари обмінюються додатків (рис. 1.3).

Такий підхід хороший для невеликої кількості додатків. При великій їх числі він практично не працює. Крім того, він не дозволяє будувати якісно нові запити до об'єднаним даними, тобто якісного виграшу від об'єднання даних немає.

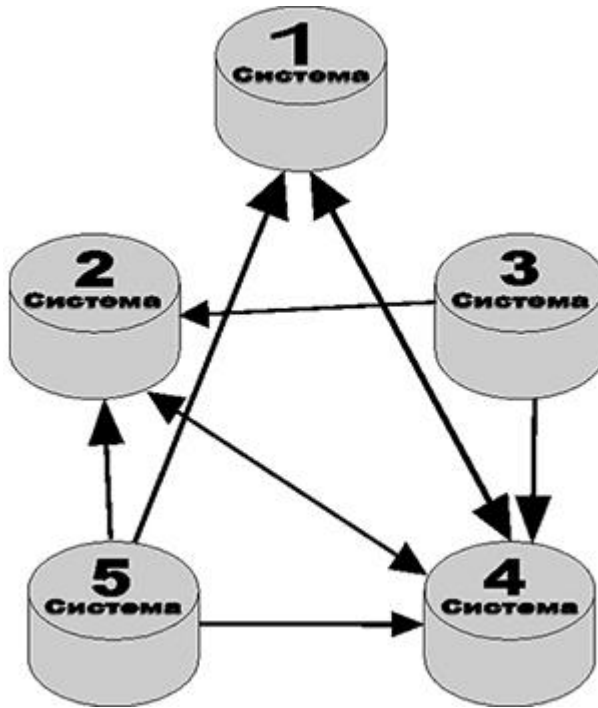


Рис. 1.3. Інтеграція "кожен з кожним"

При використанні даного підходу ми маємо N систем, оскільки кожна система має зв'язок з декількома системами (від 1 до $N - 1$), то для всієї системи в цілому необхідно створити від N до $\frac{N*(N-1)}{2}$ пар перетворень даних виду « $W_i - W_j$ », « $W_j - W_i$ »

Інтеграція за допомогою модуля інтегрування (Сервісна Шина Додатка)

Даний підхід передбачає наявність сполучної ланки між усіма модулями - Сервісна Шина Додатки. Вона здійснює управління всіма доступними модулями системи, відповідає за взаємодію між модулями, конвертування даних (якщо це необхідно), контроль якості виконаної роботи (рис 1.4).

Використання подібного підходу дозволяє мінімізувати витрати на додавання нових модулів і модернізацію поточних, знижує витрати на обслуговування системи в цілому, а також спрощує управління потоками даних у системі.

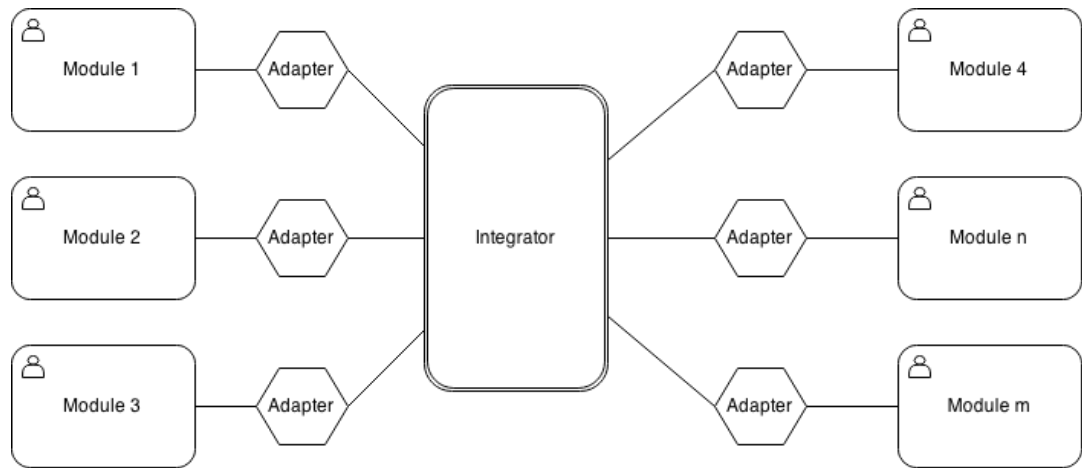


Рис. 1.4. Інтеграція за допомогою модуля інтеграції

Типова задача інтеграції

Взаємодія двох і більше прикладних систем за допомогою обміну даних.

Необхідні компоненти для інтеграції

- Адаптери прикладних систем до Сніп для відправки і прийому повідомлень;
- СШД, яка повинна складатися мінімум з:
 - Транспортна система (MQ) ;
 - Перетворення формату і даних повідомлення;
- Маршрутизація (контентна) повідомлень між прикладними системами.

Стандартні бізнес-вимоги

- Гарантована і однократна доставка повідомлень;
- Надійність роботи при мережевих, програмних і апаратних збоях (відмовостійкість);
- Висока продуктивність;
- Короткий цикл розробки інтеграційного проекту;
- Можливість легкого нарощування та повторного використання компонентів системи;
- Низька вартість володіння та обслуговування системи.

Стандартизація

- В якості стандарту тіла повідомлення приймаємо стандарт XML.
- В якості стандарту опису інтерфейсу системи приймаємо стандарт XSD.

- В якості стандарту модифікації XML повідомлень приймаємо стандарт XSLT.
- В якості мови доступу до тіла повідомлення беремо мову XPATH.

Дані стандарти добре налагоджені і мають широку підтримку.

На практиці не всі системи підтримують XML-інтерфейс для обміну даними. У цьому випадку необхідно застосовувати Сніп -Послуги перетворення формату повідомлень в XML-повідомлення (рис. 1.5).



Рис. 1.5. Обмін повідомленнями

У разі якщо формати C1 і C2 - не XML, перетворювачі можуть бути специфічним програмним продуктом. Але навіть у цьому випадку, перетворення з XML в простий нескладний формат (наприклад, comma delimited) можна виконати за допомогою XSLT-перетворення. Дії з перетворення форматів можна легко винести на рівень СШД, дані що надходять в СШД будуть перетворюватися в проміжний XML формат, а на виході будуть перетворюватися в формат системи одержувача. Самі прикладні системи не доведеться доопрацьовувати в зв'язку з цим, тому всі перетворення виконуються на рівні СШД [42].

Якщо форматом є XML, перетворення можна виконати за допомогою XSLT перетворення.

Найпоширенішим способом передачі була і є досі робота через файли. Деякі системи для взаємодії роблять наступне: робиться дамپ даних з C1, файл транспортується копіюванням, поштою або через носій і вантажиться в C2,

причому всі операції виконуються вручну. Такий процес не задовольняє вимогам сучасного бізнесу [21].

Слід мати в наявності файлові адаптери, які будуть підхоплювати файли з деякого каталогу і надсилати їх в якості повідомлень в чергу сервера повідомлень Сніп. Також вони повинні налаштовуватися на прийом повідомлень з черги сервера повідомлень. Кращим варіантом є інваріантні до ОС адаптери, для цього відмінно підходять адаптери, написані на JAVA. Інваріантність програмного забезпечення дуже важлива, так як існує багато продуктів, які працюють не тільки на ОС Windows, але і на Linux, AIX, Sun Solaris та інших. За великим рахунком, не важливо, яка ОС потрібно для роботи сервера повідомлень [30].

Багатокомпонентна інтеграція

Коли систем стає більше трьох, для їх інтеграції вже стає не вигідним використання конвертерів, які виробляють переклад з формату першої системи у формат другої системи. Системи можуть обмінюватися різними документами, для кожного з них може знадобитися унікальний перетворювач до формату тієї системи, яка повинна буде використовувати цей документ. Так, наприклад, АБС може видавати 40 видів документів, 15 можуть надходити в одну систему, 15 в другу, 10 в обидві. У такому випадку треба писати $15 + 15 + 10 * 2 = 50$ перетворювачів. І це тільки для передачі документів в одну сторону, а адже найчастіше з'являються відповідні документи, які треба також перетворювати, тільки вже у зворотний бік [43]. У разі використання проміжного формату документів при такому варіанті отримуємо 40 перетворювачів в проміжний XML-формат і по 25 перетворювачів з проміжного мови в інтерфейс систем, разом 90 перетворень. Для чого тоді так робити, якщо необхідне число перетворень стає більше? Причини наступні:

- Загальний формат документа легко описати зрозумілим для розробників мовою XSD. Можливо, є можливість доопрацювати інтеграційні компоненти таким чином, щоб вихідні документи відповідали цьому стандарту;
- За загальний формат може братися стандарт найбільшого компонента в інтеграції, в банках це зазвичай АБС, якщо банк має XML інтерфейс для

обміну даними, то це зовсім добре. Або за загальний формат може бути взятий будь-якої добре поширений і підтримуваний промисловий стандарт. Це вже буде залежати від галузі бізнесу;

- Документ, що зберігається в проміжному форматі, повинен бути легко конвертованим у формати інших систем. Ідеальним форматом таких документів є XML. Перетворення XML документів здійснюється за допомогою XSLT перетворення;
- При додаванні нової системи або заміні вже наявної, необхідно буде описати конвертер формату документа цієї системи в / з проміжного формату. Це значно економніше, ніж писати окремі перетворювачі до кожної системи при інтеграції.

Ідеальним варіантом є такий випадок:

- Всі використовувані системи мають XML-інтерфейс для взаємодії.
- За проміжний формат узятий формат найбільшого компонента (наприклад, АБС).

У такому випадку маємо:

- Число необхідних перетворень документів мінімально. Якщо зовнішній системі потрібен документ або вона відправляє свій документ в СШД, ставиться XSLT-перетворення документа до загального формату. Для описаного вище прикладу знадобитися 50 XSLT-перетворень, і при додаванні зовнішньої системи, необхідно буде додавати по одному перетворенню на кожен документ;
- Загальний формат можна розширювати додатковими документами, які ходять між системами. Для цього слід описати новий документ за допомогою XSD- схеми;
- Всі необхідні перетворення можна винести на рівень СШД, тоді прикладні системи не доведеться доопрацьовувати (доопрацювання прикладних систем може дуже затягнутися і бути досить дорогою).

Але не всі випадки ідеальні, деякі системи не вміють спілкуватися за допомогою XML і пропонують обмінюватися бінарними файлами, і всім іншим системам доводиться вчитися розуміти дані формати. У такому випадку

рішенням буде перетворення такого формату в XML для більш зручної роботи з ним. Слід також створити XSD опис отриманого формату. Перетворення можна виконати у вигляді спеціально написаного сервісу на рівні СШД, реалізація може змінюватися в залежності від використовуваного СШД. Також і у зворотний бік, якщо система приймає тільки документи у своєму специфічному форматі, СШД буде писати перетворення, яке також найзручніше винести на рівень Сніп. Причому при використанні проміжного формату документів, таке перетворення треба буде зробити один раз, а не писати конвертери, кількість яких дорівнює кількості систем, які можуть передавати цей документ.

Інтеграція на рівні інтерфейсів користувача

Підхід заснований на тому, що додатки можуть використовувати одного так само, як їх використовують люди, а саме (за допомогою спеціальних інструментів) через інтерфейс користувача (screen scraping) [52].

Інтеграція на рівні даних

Один з найпоширеніших в даний час підходів - створення сховищ даних (datawarehouses). Передбачає підтримку даних у спеціальних сховищах незалежно від бізнес-логіки, їх породила. Доступ до сховищ можуть отримувати різні додатки. При цьому підході дуже важливе значення приймає наявність добре документованої (і рідко змінюється) моделі даних [55].

Є у нього і свої недоліки, пов'язані в першу чергу з:

- підвищеними вимогами (а значить, вартістю рішення) до апаратного забезпечення серверів сховища;
- необхідністю переробляти істотну (всю роботу з БД) частина додатків (отже, збільшення вартості і можливість внесення помилок);
- необхідністю єдиної схеми даних для різномірних додатків, що істотно обмежує гнучкість і розвиток систем.

Зазвичай розглядаються два методи інтеграції даних в єдиному інформаційному процесі:

- Метод статичної інтеграції даних;
- Метод динамічної інтеграції даних.

Метод статичної інтеграції даних полягає у зв'язуванні і вбудовуванні даних одного типу, оброблюваних певним програмним засобом, в результуючий файл іншого програмного засобу. При цьому отримання даних реалізується у вигляді посилання на результат виконаної команди. Вплив системи на хід виконання команди в методі статичної інтеграції даних не допускається. При статичній інтеграції даних система оперує конкретними параметрами об'єктів і реалізує зв'язок тільки між параметрами об'єктів. При цьому в системі параметр будь-якого об'єкта зазвичай відділений від команди і є самостійною частиною об'єкта, що не залежить від будь-якої команди [49].

Метод динамічної інтеграції даних полягає у зв'язуванні однієї команди або групи команд одного програмного засобу, результатом виконання яких є значення певного типу даних, з командами іншого програмного засобу в єдиному інформаційному процесі. Причому обидва розглянутих програмних кошти мають рівні права. Порядок взаємодії визначається умовами інформаційного процесу та вимог користувача. При динамічній інтеграції даних система оперує командами, що формують параметри об'єктів, і реалізує зв'язок об'єктів безпосередньо між командами, надаючи при цьому більш гнучкий спосіб суміщення і «розуміння» даних різних типів. При цьому в системі параметр будь-якого об'єкта не повинен бути відділений від команди. Параметр є тільки формальним поданням даних в системі [51].

Розглянемо деякі САПР з позиції повноти використання в них методів динамічної та статичної інтеграції.

Безліч відомих САПР, наприклад Компас - 3D, реалізують автоматизацію проектування на рівні операцій з декларативними даними і спрощення виконання елементарних дій. Робота проектувальника в такому випадку автоматизується тільки частково, оскільки автоматизація роботи з процедурними даними залишається поза сферою можливостей САПР. Процедурні дані проекту відображаються тільки в протоколі подій, який не надає проектувальнику можливості управляти процедурними даними проекту. Оскільки процес проектування без обробки процедурних даних не обходиться, то це завдання цілком і повністю лягає на людину. В САПР даного типу ні статична, ні

динамічна інтеграція даних не реалізована і виконується проектувальником вручну [34].

В САПР з параметризацією об'єктів зв'язку між даними мають більш складну структуру. Метод параметризації даних полягає у зв'язуванні даних декларативного подання між собою за допомогою певних функцій. Результатом параметризації є побудова ланцюжка пов'язаних між собою об'єктів. Такий спосіб побудови об'єктів представляє проектувальнику більш широкі можливості управління проектом і модифікації проектного об'єкта.

САПР, що використовують параметризацію даних, мають у своїй архітектурі монітор, що відстежує стан проекту. Монітор відображає процес створення об'єкта і дозволяє здійснювати перестроювання пов'язаних об'єктів після внесення змін. Якщо в процесі проектування виникає необхідність змінити дані в одному з елементів ланцюжка зв'язаних об'єктів, то монітор проводить автоматичне перестроювання пов'язаних з ним подальших об'єктів. Застосування монітора в САПР надає проектувальнику новий потужний механізм, що дозволяє не тільки відслідковувати вироблені дії, як, наприклад, у протоколі подій, а й управляти проектом [63].

Можливості параметризації об'єктів в системі дозволяють отримати ефект відстеження стану об'єктів. Проектувальник в будь-який момент часу має можливість внести зміни в один з об'єктів проекту, не турбуючись про пов'язані з ним об'єктах. Завдяки параметризації об'єктів і контролюючому монітору, система сама виробляє перестроювання всіх пов'язаних об'єктів.

Принцип параметризації об'єктів використовується в таких САПР як DiaLux, Altium Designer. Кожен елементарний об'єкт має набір параметрів, що характеризують його стан: розмір, матеріал і т.д., що представляють собою декларативні дані. Всі елементарні об'єкти мають засоби зв'язку з такими ж об'єктами. Завдяки цим засобам зв'язку всі об'єкти можуть бути об'єднані в так звану "збірку" [41].

Засіб зв'язку об'єктів являє собою елементарну функцію зі стандартного набору функцій системи. Функції зв'язку об'єктів розрізняються за типами геометричного перекриття об'єктів один одним. Розрізняються функції, що

реалізують з'єднання в точці, з'єднання по лінії, з'єднання по поверхні і т. д. Розходження цих функцій полягає в кількості порівнюваних параметрів з'єднуються об'єктів. Залежно від результату порівняння монітором реалізуються певні дії.

Побудова елементарного об'єкта в САПР DiaLux проводиться за допомогою завдання певної поверхні і подальшим її зміною за допомогою різних графічних команд.

Наприклад, при побудові паралелепіпеда спочатку задається контур у вигляді прямокутника, після цього виконується певна операція розтягування, вдавнення, витягування і т. д., пов'язана з цим прямокутником. Специфіка даного способу зв'язку не дозволяє використовувати його для даних іншого аспекту уявлення крім графічного. У цій та подібних САПР зроблені певні кроки для вирішення даної проблеми [36].

САПР Altium Designer надає користувачеві можливість підключення різних програмних засобів за допомогою стандартних засобів зв'язку Windows. Таким чином, користувачеві надається можливість формування власної бази даних параметрів об'єкта, і використовувати їх у процесі проектування. За допомогою засобів DDE / OLE поля бази даних підключаються до параметрів об'єктів і в той момент, коли йде формування параметра, дані з бази даних заносяться в певний параметр. Оскільки процес формування баз даних, що надається САПР Altium Designer, є зовнішнім процесом, параметризація на нього не поширюється. Внаслідок цього переваги параметризованих об'єктів, що надаються САПР Altium Designer для графічних даних, не доступні для табличних даних.

Основний принцип побудови об'єктів у системі базується на операціях з декларативними даними об'єкта. Основою процесу проектування в САПР Altium Designer є послідовне виконання команд, параметри яких поєднуються з параметрами проєктованих об'єктів. Розглядаючи принципи побудови об'єктів і принципи ведення проєктів, реалізовані в САПР з параметризацією об'єктів, можна встановити, що параметри об'єктів і виконувані в процесі проектування команди відокремлені один від одного. Один і той же параметр може бути

отриманий за допомогою різних команд САПР. Параметр не є тільки формальним представленням об'єкта в системі. Перелік параметрів власне і задає об'єкт в системі. Після виконання будь-якої команди отриманий результат заноситься в перелік параметрів об'єкта. Далі при виконанні якої-небудь команди параметр оброблюваного об'єкта поєднується з параметром команди. Результат заноситься в перелік параметрів об'єкта і так далі [53].

Розглянувши всі вище перераховані властивості САПР з параметризацією об'єктів, можна зробити висновок про наявність в цих САПР інтеграції даних. Однак у таких САПР параметри об'єктів і команди відокремлені один від одного, параметри об'єктів мають пріоритет перед командами і не є тільки формальним поданням команд в системі. Внаслідок цього в САПР з параметризацією даних можуть бути реалізовані тільки методи статичної інтеграції даних. Такі САПР мають монітор, який забезпечує відстеження стан параметрів та продукує автоматичне перестроювання об'єктів.

У всіх розглянутих вище САПР немає умов для реалізації динамічної інтеграції даних. Однак програмні засоби цих САПР мають відкриту (або частково відкриту) архітектуру і можливість зв'язування з різними програмними засобами. Реалізувати динамічну інтеграцію даних можливо, створивши спеціалізовану надбудову над програмними засобами, що виконує зв'язок між командами, що створюють проектований об'єкт [62].

Комплекс програмних засобів, що мають надбудови, які реалізують динамічну інтеграцію даних, являє собою середовище автоматизованого проектування, в якій присутні програмні засоби, що обробляють дані різних аспектів уявлення. Прикладом такого середовища є інтегрована середовище САПР.

Основу методів управління процедурними множинами інтегрованого середовища САПР становить моніторинг проектних подій відповідно до інтерпретацією описів проектних процесів (сценаріїв проектування). В результаті проектні події відображаються в послідовності програмних процедур інформаційних ресурсів, що забезпечує інтегровані перетворення над описами об'єктів [39].

У розглянутій інтегрованому середовищі САПР основу методів створення прикладних користувальницьких додатків становить протоколювання процедурних подій. Протокол використовується для побудови узагальнених ланцюгів подій у вигляді автоматизованих проектних процедур, які структуруються в системній базі даних і зв'язуються з описами проектних процесів. Реалізована можливість побудови виконуваного коду проектних процедур на основі компіляції процедурних бібліотек середовища. Це дозволяє створювати програмні компоненти прикладних систем і розширювати номенклатуру уніфікованих проектних операцій [34].

Інтегроване середовище САПР дозволяє вирішувати наступні завдання:

- Виділення операцій управління перетвореннями в окремий програмний компонент - ядро середовища, що реалізує функції: управління інструментарієм середовища; управління ресурсами і периферією комп'ютера; управління інформаційним обміном; діалогового обміну з користувачем;
- Об'єднання операцій над одним типом інформаційного опису в єдину програмну систему, складову інструментальний фонд середовища;
- Підтримки засобів, що реалізують операції об'єднання і скріплення фрагментів інформаційних описів;
- Підтримки комплексу засобів, що забезпечують розширення середовища.

Повнота інформаційних описів об'єкта в інтегрованому середовищі САПР забезпечується безліччю інформаційних процесів, що реалізують операції над:

- Графічним аспектом описів об'єкта - графічний процесор;
- Табличним аспектом описів об'єкта - табличний процесор;
- Текстовим аспектом описів об'єкта - текстовий процесор.

Розглянута інтегрована середовище САПР має такі властивості:

- Повнота і цілісність описів об'єкта проектування, що забезпечується інтегрованими перетвореннями;
- Простота і зручність операцій створення опису проектних процедур, які забезпечуються складанням описів з безлічі альтернатив - інтегрованих проектних операцій;

- Гнучкість проектних процесів, що забезпечується гнучкою структурою описів проектних процедур, що дозволяє управляти переходами за сценарієм проектування та змінювати зміст реакцій на можливі події;
- Різноманітність класів проектних операцій відповідно до рівня складності проектних завдань і кваліфікацією кінцевого користувача, забезпечуваних комбінуванням проектних операцій та їх використанням як єдиного цілого;
- Простота і зручність операцій зі спільної обробці графічних, текстових і табличних описів об'єкта;
- Підтримка і суміщення об'єктно-орієнтованих і суб'єктно орієнтованих описів проектних процесів з можливістю підключення описів під процесів;
- Еволюційність розвитку, що забезпечується зворотним зв'язком на основі протоколювання дій користувача спільно з використовуваними даними та їх подальшого структурування;
- Накопичення набутих знань для подальшого синтезу виконуваних елементів, що дозволяє еволюційно розвивати систему і налаштовувати її на різні класи об'єктів проектування;
- Простота і зручність управління діалоговим взаємодією, що забезпечуються уніфікованими операціями діалогової взаємодії ядра середовища і текстовим описом діалогових процедур.

1.5 Постановка задачі

На сьогоднішній день існує безліч САПР

$$A = \cup A_i, \text{ где } i = 1, n, \quad (1.4)$$

Кожна САПР має безліч властивостей

$$B_i = \cup B_j, \text{ где } j = 1, m, \quad (1.5)$$

Необхідно створити систему, яка об'єднає в собі всі необхідні властивості використовуваних САПР:

$$V^* = \cup_{i=1}^n V_i \text{ где } V_i \in B_i, \quad (1.6)$$

і є множиною необхідних властивостей деякої САПР.

Така система повинна відповідати наступного набору критеріїв:

- Оптимальне використання обчислювальних ресурсів;
- Мінімальний час обробки команди;
- Інтерфейс повинен бути зрозумілий і зручний у використанні.

2 ДИНАМІЧНА ІНТЕГРАЦІЯ ДАНИХ

У цьому розділі пропонується інформаційна модель середовища автоматизованого проектування, що моделює поведінку в процесі проектування всіх програмних засобів, що входять до складу середовища автоматизованого проектування. Пропонується реляційний метод динамічної інтеграції даних, за допомогою якого досягається динамічна інтеграція даних різних аспектів уявлення, що використовуються в середовищі автоматизованого проектування. За допомогою мереж Петрі демонструється застосування реляційного методу динамічної інтеграції даних в середовищі автоматизованого проектування.

2.1 Інформаційна модель середовища автоматизованого проектування

Традиційно склалося так, що під аббревіатурою САПР маються на увазі графічні системи автоматизованого проектування. Розробка графіки автоматизованих першою. Зараз намітилася тенденція розширення сенсу терміну САПР. Тепер це - набір різних програмних засобів, де графічні системи застосовуються нарівні з іншими. Тому вводиться робочий термін «середовище автоматизованого проектування», яким позначається комплекс програмних засобів, що взаємодіють в процесі автоматизованого проектування.

Процес проектування завжди є надбудовою певних дій над заздалегідь визначеними параметрами із заданими граничними умовами. Причому це твердження вірне, як на початковому, так і на кінцевому етапі проектування. Будь-який процес проектування починається з обробки проектного завдання, на параметрах якого базується початковий етап створення об'єкта задовольняє граничними умовами, закладеними у завданні [53].

Розглянемо процес створення якогось абстрактного проекту. Завдання на проектування представляється у вигляді наступних складових:

- у вигляді таблиці задані дані проекту, які є початковими параметрами, і дані є граничними умовами;
- кресленнями та схемами задаються графічний опис місце програми або застосування проекту;

- текстовим описом задаються суб'єктивні та логічні характеристики проекту.

На початковому етапі проектування проводиться обробка проектного завдання. Проектувальник вибирає пріоритетний спосіб опису проекту (графічний, текстовий, табличний або розрахунково-аналітичний). Зазвичай пріоритетним вибирається графічний спосіб опису проекту. Це означає, що в процесі проектування перші дії на кожному етапі проектування будуть проводитися над графічними даними проекту. Далі в залежності від поставленого завдання виробляються дії над даними інших аспектів уявлення.

Наступним етапом проектування є безпосереднє зміна проектних даних. Виробляється зміна графічного зображення проекту. Наприклад, додається один об'єкт. Тепер необхідно уточнити правильність установки нового об'єкта. Наприклад, якщо цей об'єкт є стандартним, то необхідно уточнити його типорозміри в довідковій таблиці і провести відповідні зміни на кресленні. Далі необхідно зробити запис про встановлення нового об'єкта в пояснювальній записці до проекту і вказати деякі його характеристики [54]. Тепер потрібно прив'язати новий об'єкт до існуючих об'єктів і проектним завданням. Для цього потрібно виконати деякі розрахунки в залежності від складності і призначення об'єкта, що змінюють форму і властивості самого об'єкта або місце його встановлення. Виконані розрахунки записуються в пояснювальну записку. Далі виробляється адаптація зміненого об'єкта до умов проекту.

Припустимо, необхідно змінити об'єкт, що знаходиться в тісному зв'язку з іншими об'єктами. Після зміни деяких характеристик цього об'єкта проектувальнику для відновлення логічної зв'язаності об'єктів необхідно буде виконати всі перераховані вище операції для кожного об'єкта пов'язаного зі зміненим [59]. Якщо в проекті є безліч пов'язаних один з одним об'єктів, то проектувальнику практично доводиться повторно виконувати проектування, враховуючи всі зв'язки.

Динамічна інтеграція даних різних аспектів уявлення реалізується в ході автоматизованого проектування керуючими впливами на елементи середовища автоматизованого проектування [52]. Пропонований в дисертації метод

динамічної інтеграції даних є методом моделювання взаємодії компонентів середовища автоматизованого проектування, тобто метод динамічної інтеграції даних - це частина інформаційної моделі середовища автоматизованого проектування.

Пропонується інформаційна модель середовища автоматизованого проектування, що складається з Керуючого процесора (КП), тематичних співпроцесорів і виконуючих процесорів.

КП керує поведінкою всіх тематичних співпроцесорів, що реалізують динамічну інтеграцію даних в середовищі автоматизованого проектування. До тематичних співпроцесорами відносяться: Графічний співпроцесор (ГП), табличний співпроцесор (ТБП), Текстовий співпроцесор (ТП) і Математичний співпроцесор (МП). Тематичні співпроцесори отримують і обробляють команди від КП і передають їх виконуючим процесорам. Виконуючими процесорами є програмні засоби, що реалізують роботу з даними одного аспекту подання. До них, наприклад, відносяться AutoCAD, SolidWorks, dBase, Paradox, MS Word, MathCAD і т.д. Дана інформаційна модель відтворює реальні взаємодії програмних засобів в моделюється середовищі автоматизованого проектування (рис. 2.1).

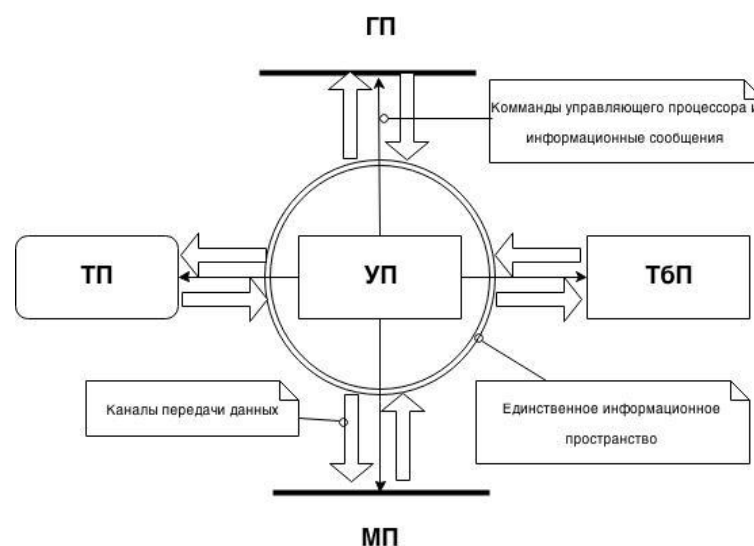


Рис. 2.1. Схема управління процесорами в інформаційній моделі

Кожен тематичний співпроцесор виконує певні дії:

- графічний співпроцесор автоматизує процедури обробки інформації, представленої у вигляді графічних зображень і підтримує відповідні структури даних для її збереження;
- табличний співпроцесор автоматизує процедури обробки інформації, представленої у вигляді таблиць і підтримує відповідні структури даних для її збереження;
- текстовий співпроцесор автоматизує процедури обробки інформації, представленої у вигляді текстових описів і підтримує відповідні структури даних для її збереження;
- математично співпроцесор автоматизує процедури розрахунку необхідних параметрів і підтримує відповідні структури даних для її збереження;
- керуючий процесор автоматизує процедури з управління процесом проектування і здійснює координацію взаємодії всіх тематичних співпроцесорів, що входять до складу середовища автоматизованого проектування.

Процес проектування являє собою виконання певних дій над даними, вираженими різними аспектами уявлення. Для досягнення приватного проектного рішення потрібно зробити кілька дій (або команд САПР). Отже, набір команд, виконуваних для отримання приватного проектного рішення, являє собою комплексну операцію. У процесі проектування є схожі комплексні операції, що мають однаковий набір команд і відрізняються тільки використовуваними даними.

Вводяться поняття:

- проект - пов'язане опис об'єкта проектування представлене у вигляді виконаних узагальнених операцій в послідовності, рекомендованої сценарієм проектування. До складу проекту також входять окремі команди тематичних співпроцесорів не входять до складу узагальнених операцій. Проект виконується вручну проектувальником з використанням узагальнених операцій із складу бібліотек операцій.
- узагальнена операція - набір команд для одного або кількох тематичних співпроцесорів. Команди співпроцесора є процедурної складової (Т)

проектних даних, пов'язаних між собою операторами зв'язку К. Узагальнена операція призначена для виконання певної послідовності дій, заданої проектувальником. Узагальнені операції можуть бути пов'язані між собою за допомогою операторів К, що здійснюють зв'язок між окремими командами з різних узагальнених операцій.

- сценарій проектування - рекомендований проектувальнику набір пов'язаних між собою узагальнених операцій. Сценарій проектування задає послідовність виконання узагальнених операцій при формуванні закінченого проекту.
- монітор - засіб зі складу КП, що відстежує стан проекту. У моніторі відбивається стан проекту у вигляді послідовності виконаних команд тематичних процесорів. Монітор відображає виконані команди і зв'язки між ними, дозволяє редагувати виконані узагальнені операції і перевизначити зв'язок між командами.

Дана інформаційна модель надає можливість створювати і відпрацьовувати сценарій проектування, що включає в себе узагальнені операції, що виконуються в процесі проектування. Результатом узагальненої операції є виконання конкретного етапу проектування. Узагальнена операція може складатися з команд як одного, так і декількох тематичних процесорів. Наприклад, в процесі проектування потрібно відкрити довідкові таблиці, звідки отримані дані для конкретного об'єкта. У цій узагальненій операції виконуються наступні дії:

- визначається тип об'єкту;
- визначається ім'я файлу бази даних містить довідник;
- визначається параметри для пошуку даних за ключовим словом;
- проводиться пошук необхідної записи в базі даних для конкретного об'єкта.

Використання узагальненої операції значно спрощує цю процедуру, і тепер проектувальнику не обов'язково знати, де і як зберігаються використовувані ним дані. Завдяки узагальненій операції система сама знає всі необхідні відомості технічного характеру і видає результат тільки по одній команді проектувальника.

У сценарії проектування відбивається вся послідовність дій виконувана проектувальником. Наявність сценарію дозволяє звести до мінімуму виконання повторюваних дій. Наприклад, необхідно розмістити і описати деяку кількість подібних об'єктів. Для цього проектувальник один раз проходить всю послідовність дій, використовуючи весь арсенал наявних узагальнених операцій, і позначивши цю частину сценарію, може пройти її знову, змінюючи тільки вхідні параметри.

Сценарій проектування може мати деревоподібну структуру, де гілками є узагальнені операції. У точках з'єднання гілок управління передається проектувальнику. Можливі різні варіанти реалізації сценарію проектування. З одного проектного стану сценарій може бути продовжений різними узагальненими операціями. Або ж у сценарії одне і теж дія може бути реалізовано декількома узагальненими операціями, наприклад, якщо є кілька методик розрахунку.

При формуванні проекту проектувальник працює з КП, за допомогою якого вибирає з бібліотеки відповідний сценарій проектування і запускає його на виконання. КП виконує узагальнені операції сценарію, формуючи проект. При цьому проектувальник аналізує формовані декларативні дані Р і веде діалог з КП. У найпростішому випадку створення проекту на цьому завершується. У загальному випадку за результатами аналізу проектувальник доповнює проект окремими командами або узагальненими операціями, налаштовуючи запропонований сценарій для конкретної проектної ситуації. КП надає проектувальнику можливість формувати власні узагальнені операції, створювати власні сценарії проектування або редагувати існуючі. Формування узагальнених операцій проводиться в моніторі, де з окремо виконаних вручну команд збираються узагальнені операції і визначаються оператори зв'язку До між командами, які є процедурними даними Т. Тематичні співпроцесори реалізують прямі і зворотні зв'язки ко Анд формованого проекту (окремих або з узагальнених операцій) з виконуючими процесорами. Динамічна інтеграція в даній інформаційній моделі середовища автоматизованого проектування

реалізується в загальному випадку керуючими впливами проектувальника в процесі діалогу з КП.

Керуючий процесор в інформаційній моделі середовища автоматизованого проектування грає головну роль. За допомогою нього відпрацьовується сценарій проектування. На основі сценарію Керуючий процесор вирішує які команди подаються тематичним співпроцесорами для виконання тієї чи іншої узагальненої операції наявної в сценарії проектування.

Для найбільш якісного відпрацювання сценарію проектування Керуючим процесором ведеться протокол виконуваних дій, завдяки якому проектувальник в будь-який момент часу має можливість повернутися назад і виконати всі пройдені дії знову.

Протокол не дублює сценарій проектування. Сценарій надає проектувальнику можливість виконувати узагальнені операції в тій послідовності, в якій дії призведуть до кінцевого результату. Але в сценарії проектування не відомо скільки, наприклад, раз були виконані ті самі узагальнені операції для подібних об'єктів. Ця можливість надається тільки протоколом подій.

Кожен з тематичних співпроцесорів обробляє дані тільки того аспекту вистави для операцій, з якими він призначений. Якщо, йому будуть потрібні дані, що обробляються іншим тематичним співпроцесором, то він подає інформаційне повідомлення Керуючому процесору. Той, у свою чергу, перериває роботу першого співпроцесора і дає команду відповідному співпроцесору надати необхідні дані в єдиному інформаційному просторі. Після отримання повідомлення про готовність даних Керуючий процесор подає команду першому співпроцесору продовжувати роботу. Тематичні співпроцесори не можуть звертатися до таких же співпроцесорами безпосередньо, для цих цілей використовується Керуючий процесор.

Тематичні співпроцесори можуть подавати тільки інформаційні повідомлення, і отримувати тільки команди. Керуючий процесор отримує тільки повідомлення і видає тільки команди. Тематичні співпроцесори, звертаючись до Керуючому процесору, отримують від нього команду на переривання роботи, і

продовжують свою роботу тільки після отримання відповідної команди Керуючого процесора. Використання такої моделі Управління дозволяє здійснювати контроль над виробленими діями (рис. 2.2).

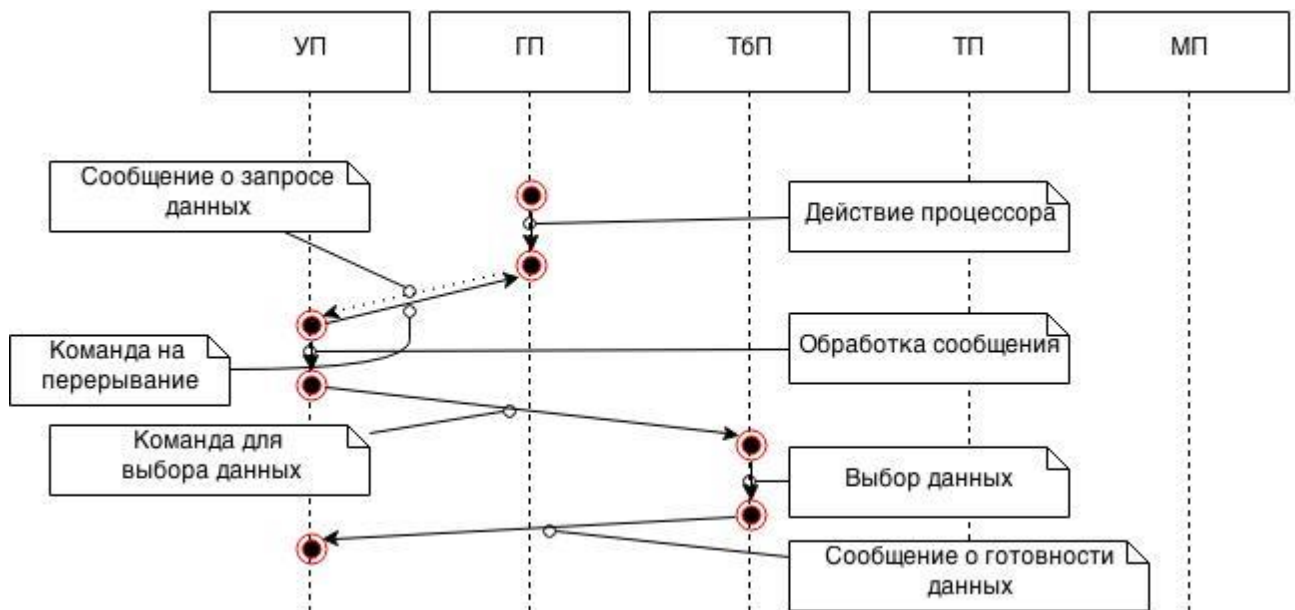


Рис. 2.2. Схема взаємодії тематичних співпроцесорів і КП

Кожен співпроцесор в будь-який момент часу надає проектувальнику можливість коригувати процес виконання дій з сценарію проектування. Тематичні співпроцесори мають ряд властивостей і функцій сервісного характеру. Вони керують процесом видачі інформаційних повідомлень і реагують на команди Керуючого процесора. Ці функції є особливо важливими для підтримки динамічної інтеграції всіх процесів системи.

Однією з головних можливостей тематичних співпроцесорів є реєстрація всіх дій проектувальника. Завдяки цьому стає можливим створення власних узагальнених операцій і створення певних етапів сценарію проектування, формованих проектувальником на свій розсуд. Для підтримки взаємодії всіх процесів, що відбуваються в середовищі автоматизованого проектування, тематичні співпроцесори використовують команди обробки та трансформації параметрів. Параметри перетворюються до виду зрозумілому кожному тематичному співпроцесору включеного в середу автоматизованого проектування, і передаються в єдиний інформаційний простір. Зворотне перетворення даних отриманих з єдиного інформаційного простору дозволяє проводити дії над об'єктами відповідно з даними, отриманими від інших

тематичних співпроцесорів. Для підтримання взаємодії один з одним тематичні співпроцесори мають команди передачі оброблених даних в єдиний інформаційний простір і отримання з нього даних, прийнятих від інших співпроцесорів. Кожен тематичний співпроцесор призначений тільки для виконання своїх, суворо визначених, дій.

Графічний співпроцесор виконує дії лише над графічними зображеннями. У процесі проектування графічний співпроцесор займає значне місце, оскільки креслення є важливою складовою будь-якого проектного рішення. Графічний процесор дозволяє виконувати креслення і виробляти редагування наявних графічних об'єктів.

Табличний співпроцесор виконує дії лише над табличними даними. У процесі проектування іноді важко обійтися без табличних даних, оскільки таблиця є зручною формою розміщення великої кількості однотипних даних. Табличний співпроцесор реалізує табличні дані, створюючи і використовуючи бази даних. Проектувальник постійно звертається до довідкових даних різного характеру, які часто представлені у табличній формі. Одним з результатів проектування є таблиці розрахованих параметрів, таблиці обладнання, специфікації і т.д.

У процесі розробки проектної документації постійно використовується текстовий співпроцесор. Він виконує дії лише над текстовими даними. Для повного уявлення проектного об'єкта у проектній документації обов'язково наявність пояснювальної записки. Для цього проектувальнику необхідно постійно в процесі проектування створювати нові і вносити зміни в існуючі тексти.

Текстовий співпроцесор має деяку відмінність від інших тематичних співпроцесорів. Дані, створювані в текстовому процесорі, в основному носять описовий характер і тому вплив цих даних на інші співпроцесори мінімально.

В процесі створення проекту може використовуватися математичний співпроцесор, так як в проектній діяльності часто потрібні математичні розрахунки. В основному робота математичного співпроцесора проектування під час роботи невидима. Математичні формули і вирази пов'язують чисельні

значення параметрів команд різних тематичних співпроцесорів. Але в ситуаціях вимагають вибору тієї чи іншої методики розрахунку, математичний співпроцесор надає проектувальнику можливість управління математичними розрахунками.

Інформаційна модель дозволяє об'єднати Керуючі впливи проектувальника, що забезпечують динамічну інтеграцію даних, в узагальненій операції, що складається з команд різних тематичних співпроцесорів, що працюють з одним набором декларативних даних. Такі узагальнені операції (створені проектувальником або наявні в бібліотеці) дозволяють автоматизувати динамічну інтеграцію даних. Для формування узагальнених операцій необхідно виявити загальні закономірності зв'язків між процедурними даними різних (або одного) аспектів уявлення.

2.2 Представлення і взаємодія даних в інформаційній моделі

Динамічна інтеграція даних різних аспектів уявлення полягає в забезпеченні можливості обміну даними одного типу (цілого, дробового, символьного тощо) між програмними засобами, що представляють дані в різних форматах [54]. Суть методу полягає в тому, щоб зв'язати дані одного типу різних аспектів подання в потрібний момент процесу проектування відповідними операторами К.

Узагальнені операції, реалізують динамічну інтеграцію декларативних даних Р, можливі тільки при наявності операторів зв'язку К між процедурними даними Т. Оператори К об'єктивно існують в будь-якому реалізованому наборі проектних даних. Процедурні дані Г, що представляють собою команди тематичних процесорів, мають параметри, що представляють собою проектні дані - декларативні дані Р. Динамічна зв'язок між командами тематичних співпроцесорів реалізується операторами зв'язку К шляхом зв'язування параметрів команд. Таким чином, реалізувавши зв'язку процедурних даних Г, стає можливим отримати динамічну інтеграцію декларативних даних Р.

Стосунки, подібні реалізованим операторами зв'язку K , прийнято називати реляційними. Відповідно реляційний сенс мають команди тематичних співпроцесорів, що обробляють процедурні дані T .

Пропонується реляційний метод динамічної інтеграції, що полягає у використанні в процесі проектування узагальнених операцій, що складаються з команд різних тематичних співпроцесорів і інтегруючих дані як одного, так і різних аспектів уявлення.

Існують загальні закономірності зв'язків між процедурними даними різних (або одного) аспектів уявлення. Процедурними даними T є команди тематичних співпроцесорів, обробні декларативні дані P відповідних аспектів уявлення.

При динамічної інтеграції даних декларативні дані P і процедурні дані T зв'язуються в єдине опис проекту за допомогою операторів зв'язку K , тобто динамічна інтеграція даних полягає у формуванні конкретного набору K при створенні конкретного проекту.

Система описів об'єкта проектування має стани, при яких система автоматично не вирішується або має безліч рішень, що ускладнюють автоматизацію процесу проектування. Для вирішення даної проблеми потрібно ввести деякі обмеження в безліч K , що представляють собою правила формування системи.

Безліч операторів зв'язку K визначає відносини між даними одного аспекту подання або між даними різних аспектів уявлення.

Відношення діляться на:

- Транзитивні - відношення між елементами даних справжні як у прямій, так і в зворотній послідовності;
- Не транзитивні - відношення між елементами даних справжні в прямій і помилкові в зворотній послідовності;
- Виключні - відношення між елементами даних справжні в прямій і неіснуючі в зворотній послідовності.

Розглянемо можливі відношення між даними одного аспекту уявлення.

Відношення між даними графічного аспекту представлення.

Дані графічного аспекту уявлення використовуються для відображення графічної інформації. К цього аспекту даних відносяться такі властивості графічного об'єкта:

- найменування команди - декларативні дані, що визначає дані процедурного характеру;
- способи побудови - процедурні дані, що визначають послідовність виконання команди, яка визначає способи побудови графічного об'єкта;
- способи зв'язування - процедурні дані, що визначають способи зв'язування між собою декларативних даних різних графічних об'єктів.
- параметри об'єкта - всі доступні декларативні дані об'єкта, значення яких можуть змінюватися після закінчення побудови об'єкта.

Способи побудови графічних об'єктів різні. Існують об'єкти які мають:

- винятковими способами побудови, коли є тільки один спосіб побудови не залежно від кількості параметрів об'єкта (наприклад, тривимірні фігури AutoCAD);
- транзитивними способами побудови, що дозволяють довільно змінювати послідовність визначення декларативних даних об'єкта без зміни місця розташування об'єкта (наприклад, прості об'єкти AutoCAD: лінія, точка, промінь, конструкційна лінія);
- не транзитивними способами побудови, коли застосування різних способів побудови без зміни декларативних даних до одного об'єкту змінює його місце розташування (наприклад, складні об'єкти AutoCAD: дуга, коло, еліпс).

Способи зв'язування об'єктів також володіють певними умовами. Для різних графічних об'єктів застосовні певні способи зв'язування:

- виняткові способи зв'язування - побудова залежності одного об'єкта від іншого тільки певним способом зв'язування (наприклад, метод зв'язування по квадранту застосовується лише до дуги, кола, еліпсу);
- транзитивні способи зв'язування - побудова залежності одного об'єкта від іншого способом, при якому можливо змінити залежність об'єктів

протилежно, застосовуючи такий самий спосіб зв'язування (наприклад, метод прив'язки по кінцевим точкам);

- не транзитивні способи зв'язування - побудова залежності одного об'єкта від іншого способом, визначальним залежність об'єкта тільки в одному напрямку ієрархії об'єктів (наприклад, прив'язка одній лінії до центру іншого).

Існують ситуації, при яких спосіб зв'язування безпосередньо залежить від способу побудови. При певному способі побудові об'єкта можуть використовуватися додаткові дані, які після закінчення побудови не входять до складу параметрів об'єкта. Наприклад, основними параметрами окружності є координата центру і радіус. Побудувати коло можна по трьох точках, не відображати в параметрах. У таких випадках доводиться використовувати додаткові дані, до яких прив'язується побудова об'єкта.

Відношення між даними табличного аспекту представлення.

Дані табличного аспекту уявлення використовуються для відображення інформації в табличному вигляді. Табличний аспект представляється у вигляді табличного об'єкта володіє наступними властивостями:

- найменування таблиці - декларативні дані, що визначають шлях доступу до вмісту таблиці;
- рядки таблиці - декларативні та процедурні дані, що визначають шлях доступу до даних, що містяться в таблиці і спосіб їх обробки;
- елементи таблиці - декларативні та процедурні дані, що визначають шлях доступу до даних, що містяться в таблиці і спосіб їх обробки;
- операції над даними - процедурні дані, що визначають процедури і команди обробки таблиці.

Операції над даними таблиці представляють собою набір команд СУБД, що виконують певні дії при обробці таблиці. Кожна команда, що входить до складу операції, виконує певні дії над вмістом таблиці. Команди зі складу операцій мають можливість зв'язування з іншими командами, як з власної операції, так і з іншої. Відрізняються такі способи зв'язування:

- винятковий спосіб зв'язування - команда зв'язується з певним властивістю таблиці (наприклад, вказується певна рядок і комірка таблиці);
- транзитивний спосіб зв'язування - властивість однієї таблиці зв'язується з властивістю іншої, при якому можливо змінити залежність властивостей протилежно, застосовуючи такий самий спосіб зв'язування (наприклад, група осередків однієї таблиці зв'язується з групою осередків іншої таблиці);
- не транзитивний спосіб зв'язування - побудова залежності одного властивості таблиці від іншого способом, визначальним залежність властивостей тільки в одному напрямку ієрархії властивостей (наприклад, одна команда з табличній операції пов'язується з іншою операцією, для процедурних даних; і ставлення «один до багатьох» в декларативному поданні даних).

Розрізняються два основні режими роботи з таблицями: режим модифікації таблиці і режим читання даних з таблиці. У режимі модифікації таблиці проводиться запис певних даних в осередку, відповідно виконуються команди модифікують записи таблиці. У режимі читання таблиці здійснюється тільки переміщення покажчика по осередках і читання даних із зазначених осередків.

Наявність різних процедур модифікації і процедур читання ускладнює роботу з таблицями. Доцільно використовувати різні методи для читання і модифікації в одній процедурі, за аналогією з властивостями об'єктів. Одна і та ж таблична процедура має подвійну дію. У режимі читання команди, що змінюють вміст таблиці ігноруються, і команди записи замінюються командами читання даних. У режимах читання і модифікації команди, що позиціонують покажчик працюють однаково.

Застосування методів читання і модифікації накладає певну умову, що обмежують свободу формування табличних процедур, яке виражається в зменшенні транзитивних і виняткових способів зв'язування команд в рамках однієї процедури.

Відношення між даними текстового аспекту представлення.

Дані текстового аспекту уявлення використовуються для відображення інформації в текстовому вигляді. Текстовий аспект представляється у вигляді текстового об'єкта володіє наступними властивостями:

- найменування файлу - декларативні дані, що визначають шлях доступу до вмісту текстового файлу;
- вміст текстового файлу (текст) - декларативні дані, що містять в собі дані необхідні для опису процесу проектування та проектного об'єкта;
- поля і посилання - декларативні та процедурні дані, що визначають зв'язок декларативних даних текстового файлу з даними інших програмних засобів;
- команди - процедурні дані, що визначають процедури і команди текстових процесорів для формування та форматування тексту.

Включаються поля і посилання необхідні для зв'язування декларативних даних текстового аспекту вистави з декларативними даними як текстового, так і інших аспектів уявлення. Кожне включається полі або посилання на об'єкт змінює смислове значення тексту. За допомогою полів і посилань реалізується зв'язок між декларативними даними різних програмних засобів обробляють дані відповідних аспектів уявлень. Відрізняються такі способи зв'язування:

- винятковий спосіб зв'язування - в текст вставляється посилання на певний об'єкт дані якого використовуються в тексті і зміна вмісту текстового файлу не робить впливу на вміст пов'язаного об'єкта (наприклад, вставка малюнка, формули або написи);
- транзитивний спосіб зв'язування - зв'язування посилання з зовнішнім об'єктом, причому зміна даних можливе як в самому тексті, так і у вбудовується об'єкті (наприклад, зв'язок шляхом вставки об'єкта);
- не транзитивний спосіб зв'язування - зв'язування даних тексту з параметрами цього або іншого текстового файлу, коли пов'язаний текст повністю залежить від вставленого параметра (наприклад, вставка значень поточної дати, часу, параметрів документа і т.п.)

Команди текстового процесора призначені для виконання операцій над документом, текст, що включаються полями і посиланнями, а також для

підтримки зв'язку текстового процесора з іншими програмними засобами. Команди текстового процесора можуть об'єднуватися в набори для виконання певних операцій, для стислості набори команд будемо називати операціями. У деяких ситуаціях команди можуть виконуватися окремо.

Зв'язок команд текстового процесора може здійснюватися між операціями, між окремими командами, що не входять в операції, між окремими командами та операціями. Є можливість використовувати зв'язок між командами входять до складу операцій з окремими командами, з операціями або з командами входять до складу інших операцій.

Зв'язок між операціями виконується наступними способами:

- винятковий спосіб зв'язування - операція пов'язується з іншого певної операцією, причому ці операції повністю залежать один від одного (наприклад, операція відкриття макросів документа пов'язана виключно з операцією відкриття текстового документа);
- транзитивний спосіб зв'язування - зв'язок між операціями, коли виконання першої з пов'язаних операцій впливає на другу, і навпаки виконання другої впливає на першу (наприклад, операції виведення однакових текстових даних у різних частинах документа або в різних форматах);
- не транзитивний спосіб зв'язування - побудова залежності однієї операції від іншої, способом, визначальним залежність операцій тільки в одному напрямку ієрархії (наприклад, операція заповнення клітинок вбудовуваної в текст таблиці залежить від операції отримання даних).

Зв'язок між окремими командами, що не входять в операції виконується наступними способами:

- винятковий спосіб зв'язування - команда зв'язується з іншого певною командою, причому ці команди можуть зв'язуватися тільки між собою і ні з якою іншою (наприклад, команда копіювання тексту в область обміну даними і команда вставки даних);
- транзитивний спосіб зв'язування - зв'язок між командами, коли пов'язані команди можуть виконуватися в будь-якій послідовності, впливаючи на пов'язану команду (наприклад, команди форматування за зразком,

пов'язані між собою, можуть виконуватися в будь-якій послідовності, форматуючи абзаци по одному заданої);

- не транзитивний спосіб зв'язування - побудова залежності однієї команди від іншої способом, визначальним залежність команд тільки в одному напрямку ієрархії.

Зв'язок між окремими командами, що не входять до складу операції і цілими операціями виконується наступними способами:

- винятковий спосіб зв'язування - команда зв'язується з цілою операцією, причому зв'язок може бути виконана тільки між ними і ні з якими іншими командами або операціями (наприклад, операція пошуку використовуваних в відкривається документі макросів може бути пов'язана тільки з командою відкриття документа);
- транзитивний спосіб зв'язування - зв'язок між окремою командою і операцією, коли пов'язані команди і операції можуть виконуватися в будь-якій послідовності, впливаючи один на одного (наприклад, команди та операції форматування абзців за зразком);
- не транзитивний спосіб зв'язування - побудова залежності операції від команди способом, визначальним залежність операції від команди тільки в одному напрямку ієрархії.

Команди, що входять до складу операції мають можливість зв'язуватися як з окремими командами, так і з цілими операціями. При зв'язуванні команд, що входять до складу операцій застосовуються такі ж способи зв'язування, як і для окремих команд. Оскільки ці команди вже мають зв'язку з командами цієї ж операції, то на способи зв'язування цих команд накладаються деякі обмеження.

Команди, що входять до складу операції, можуть зв'язуватися підрядним зв'язком тільки при наявності вільних від зв'язків параметрів команди. Підрядний зв'язок може виконуватися тільки винятковим або не транзитивний способами зв'язку. При виконанні операція, що має в своєму складі пов'язану команду, буде виконуватися повністю лише після повного виконання іншої пов'язаної операції. Якщо команда, що входить до складу операції, зв'язується з командою, що входить до складу іншої операції, що має циклічне повторення команд, то

операція, що включає в себе пов'язану команду, буде виконуватися стільки разів, скільки інша пов'язана операція.

Відношення між даними математичного аспекту уявлення.

Дані математичного аспекту уявлення використовуються для відображення математичної інформації. За допомогою математичних виразів і формул виробляються обчислення необхідних у процесі проектування значень. До цього аспекту уявлень відносяться наступні об'єкти властивості математичного процесора:

- вхідні параметри - декларативні та процедурні дані представляють собою аргумент функції або формули;
- вихідні значення - декларативні та процедурні дані є результатом функції або формули.

Виконання математичних розрахунків являє собою підстановку необхідних значень в якості аргументів математичної функції, і отримання результатів обчислень. Для різного роду обчислень можуть використовуватися як одна формула, так і набір пов'язаних між собою виразів. Зв'язування виразів проводиться за допомогою декларативних складових аргументів і вихідних значень. Кожен параметр математичного виразу записується певним набором символів, який використовується в якості аргументу або вихідного значення залежно від призначення формули.

Способи зв'язування математичних виразів мають деяку особливість, що відрізняє їх від представлених вище способів зв'язування даних різних аспектів уявлення. Вихідні значення можуть використовуватися тільки для підстановки в якості аргументів математичних виразів. У даній ситуації відсутня транзитивний спосіб зв'язування, оскільки на першу функцію не можуть впливати обчислення, вироблені другою функцією. Таким чином, використовуються два способи зв'язування математичних об'єктів:

- винятковий спосіб зв'язування - математичний вираз своїм вихідним значенням зв'язується з аргументом іншого математичного виразу і символічне вираз, що визначає вихідне значення математичного

виразу більше ні де не використовується (наприклад, застосування проміжної функції для серії обчислень);

- не транзитивний спосіб зв'язування - аргумент одного математичного виразу пов'язаний вихідним значенням іншого, причому аргумент може використовувати вихідне значення тільки одного математичного виразу, а з одним вихідним значенням можуть бути пов'язані аргументи декількох математичних виразів.

Відношення між даними різних аспектів уявлення.

Відношення між даними різних аспектів подання визначається безліччю операторів зв'язку K. Способи відносин даних різних аспектів уявлення між собою виражаються у вигляді зв'язування процедурних складових даних. Внаслідок різнорідності застосовуваних способів зв'язування даних різних аспектів вистави для зв'язку кожної пари аспектів застосовуються персональні оператори зв'язку з безлічі K.

Для графічного аспекту подання даних можливо встановити підлеглі зв'язку з іншими аспектами представлення даних. При зв'язку з табличними даними використовуються три способи зв'язування:

- транзитивний спосіб зв'язування - зв'язування одного параметра графічної команди безпосередньо з командами вибору стовпців або рядків таблиці з табличній операції;
- не транзитивний спосіб зв'язування - зв'язування параметра графічної команди з командами вибору значень з таблиці;
- винятковий спосіб зв'язування - зв'язування команди графічного процесора з табличній операцією цілком, що дозволяє виконати графічну команду тільки після виконання табличній операції.

При зв'язку графічних команд з математичним процесором можливі тільки два способи зв'язування даних:

- не транзитивний спосіб зв'язування - зв'язування одного параметра графічної команди з командами обробки множин і матриць математичного пакета;

- винятковий спосіб зв'язування - зв'язування одного параметра графічної команди з конкретним вихідним значенням математичного виразу.

Для зв'язку графічних команд з командами текстового процесора надаються три способи зв'язку:

- транзитивний спосіб зв'язування - графічна команда зв'язується з операцією або командою текстового процесора і в даному способі зв'язку надається можливість взаємного впливу пов'язаних параметрів команд один на одного;
- не транзитивний спосіб зв'язування - графічна команда зв'язується з операцією або командою текстового процесора, причому виконання графічної команди можливо тільки після завершення виконання операції або команди текстового процесора
- винятковий спосіб зв'язування - графічна команда пов'язується тільки з певною операцією або командою текстового процесора і виконання пов'язаних команд і операцій можливо в будь-якій черговості

Для табличного аспекту подання даних можливо встановити підлеглі зв'язку з іншими аспектами представлення даних. При зв'язку з графічними даними використовуються три способи зв'язування:

- транзитивний спосіб зв'язування - зв'язування команди вибору стовпців або рядків таблиці з табличній операції безпосередньо з одним параметром графічної команди;
- не транзитивний спосіб зв'язування - зв'язування команди вставки значень в таблицю з параметром графічної команди;
- винятковий спосіб зв'язування - зв'язування операції табличного процесора дозволяє виконати табличну операцію тільки після виконання графічної операції.

При зв'язку з математичними даними використовуються тільки два способи зв'язку:

- не транзитивний спосіб зв'язування - зв'язування команди або операції вставки значень в таблицю з командою обробки матриць або множин математичного пакета;

- винятковий спосіб зв'язування - зв'язування команди або операції вставки значень в таблицю з конкретним вихідним значенням математичного виразу.

При зв'язку з текстовими даними використовуються три способи зв'язку:

- транзитивний спосіб зв'язування - таблична команда або операція пов'язується з операцією або командою текстового процесора і в даному способі зв'язку надається можливість взаємного впливу пов'язаних параметрів команд один на одного;
- не транзитивний спосіб зв'язування - таблична команда або операція пов'язується з операцією або командою текстового процесора, причому виконання графічної команди можливо тільки після завершення виконання операції або команди текстового процесора;
- винятковий спосіб зв'язування - таблична команда або операція пов'язується тільки з певною операцією або командою текстового процесора і виконання пов'язаних команд і операцій можливо в будь-якій черговості.

Для математичного аспекту подання даних можна встановити підлеглі зв'язку з іншими аспектами даних. В силу особливостей математичного аспекту подання даних для зв'язків з іншими аспектами неможливо використовувати транзитивний спосіб зв'язку. При зв'язку з графічними даними використовуються два способи зв'язування:

- не транзитивний спосіб зв'язування - зв'язування команди обробки множин з одним параметром графічної команди;
- винятковий спосіб зв'язування - зв'язування аргументу математичного виразу з конкретним параметром графічної команди.

Для зв'язку з табличними даними використовуються два способи:

- не транзитивний спосіб зв'язування - зв'язування команди обробки матриць або множин з командами вибору рядків і осередків таблиці;
- винятковий спосіб зв'язування - зв'язування аргументу математичного виразу з командою вибору значень з таблиці.

Особливість текстового процесора як засобу відображення проектних даних накладає на оператор зв'язку К між даними математичного та текстового аспекту обмеження на застосування різних способів зв'язування. Для зв'язку даних математичного аспекту уявлень з текстовим аспектом підрядним зв'язком неможливо використовувати транзитивний спосіб зв'язування. Чи не транзитивний спосіб зв'язування набуває властивостей виняткового способу зв'язування. Тому для зв'язку з текстовим аспектом подання даних використовуються тільки винятковий спосіб зв'язування - зв'язування аргументу математичного виразу з певною командою або операцією текстового процесора.

Для текстового аспекту подання даних можливо встановити підлегли зв'язку з іншими аспектами представлення даних. Для зв'язку команд і операцій текстового процесора з графічними командами надаються три способи зв'язку:

- транзитивний спосіб зв'язування - операція або команда текстового процесора зв'язується з графічною командою і в даному способі зв'язку надається можливість взаємного впливу пов'язаних параметрів команд один на одного;
- не транзитивний спосіб зв'язування - операція або команда текстового процесора пов'язується тільки з графічною командою, причому виконання пов'язаної операції або команди текстового процесора можливо тільки після завершення виконання графічної команди;
- винятковий спосіб зв'язування - текстова операція або команда пов'язується тільки з певною графічною командою і виконання пов'язаних команд і операцій можливо в будь-якій черговості.

При зв'язку з табличними даними використовуються три способи зв'язування:

- транзитивний спосіб зв'язування - зв'язування параметра однієї операції або команди текстового процесора безпосередньо з командами вибору стовпців або рядків таблиці з табличній операції;
- не транзитивний спосіб зв'язування - зв'язування параметра операції або команди текстового процесора з командами вибору значень з таблиці;

- винятковий спосіб зв'язування - зв'язування операції або команди текстового процесора з табличній операцією цілком, що дозволяє виконати операцію або команду текстового процесора тільки після виконання табличній операції.

При зв'язку операцій або команд текстового процесора з математичним процесором можливі тільки два способи зв'язування даних:

- не транзитивний спосіб зв'язування - зв'язування одного параметра операції або команди текстового процесора з командами обробки множин і матриць математичного пакета;
- винятковий спосіб зв'язування - зв'язування одного параметра операції або команди текстового процесора з конкретним вихідним значенням математичного виразу.

Безліч операторів зв'язку R складається з підмножин, що відносяться до певного аспекту подання даних.

Таким чином, кожна підмножина відносин певного аспекту подання даних ділиться на групи відносин «всередині себе» і «з усіма іншими».

За типами відносин тематичні підмножини операторів зв'язку групуються наступним чином:

- транзитивні;
- не транзитивні;
- виняткові.

Множини містять різні набори підмножин, що обумовлено принциповим відсутністю деяких видів зв'язків між даними різних аспектів уявлення.

Наведені вище формальні обмеження дозволяють автоматизувати процес контролю формованих проектувальником відносин між даними. Це дозволить зменшити витрати на динамічну інтеграцію даних, виключивши формальні помилки проектування.

У реальних множинах R існують ще й смислові обмеження на динамічну інтеграцію даних, які виявляє проектувальник. Наприклад, обмеження такого типу: «формально ніщо не заважає зобразити електродвигун в певному місці

схеми СЕС, але це в даній схемі не потрібно для її роботи». У певному сенсі робота проектувальника і полягає у відборі потрібного варіанту з усіх можливих.

Висновки до розділу

У цьому розділі пропонується інформаційна модель середовища автоматизованого проектування, що включає в себе Керуючий процесор і чотири тематичні співпроцесора: Графічний співпроцесор, табличний співпроцесор. Текстовий співпроцесор і Математичний співпроцесор. Даються поняття проекту, узагальненої операції, сценарію проектування і монітора як засобів, в інформаційній моделі середовища автоматизованого проектування. Розробляється схема і порядок взаємодії тематичних співпроцесорів і Керуючого процесора.

Розроблено способи подання та взаємодії даних в інформаційній моделі середовища автоматизованого проектування. Пропонується розділяти відношення між даними на такі види: транзитивні, не транзитивні, виняткові. Розглядаються можливі види відношень між даними як одного, так і різних аспектів уявлення. Встановлено, що не всі програмні засоби підтримують дані види відношень повністю. Пропонується використовувати отриманий набір можливих відношень як формальне обмеження для середовища автоматизованого проектування.

Пропонується реляційний метод динамічної інтеграції даних в середовищі автоматизованого проектування, що полягає у використанні в процесі проектування узагальнених операцій, що складаються з команд різних тематичних співпроцесорів і інтегруючих дані як одного, так і різних аспектів уявлення. Розглядається механізм інтерпретації реляційного методу динамічної інтеграції за допомогою мережі Петрі, що дозволяє формальними методами аналізувати процеси проектування в умовах використання узагальнених операцій.

3 СТРУКТУРНИЙ СИНТЕЗ ІНТЕГРОВАНИХ САПР

Розглянемо докладніше можливості розробки програмних засобів, що забезпечують динамічну інтеграцію даних по запропонованому реляційному методу в середовищі автоматизованого проектування. Для цього розробляється комплекс програмних засобів, що складається з керуючого процесора, тематичних співпроцесорів і виконуючих процесорів. Для зручності програмної реалізації керуючий процесор представлений як серверний вузол. Тематичні співпроцесори згруповані як засоби динамічної інтеграції даних. Окремо розглядаються допоміжні програмні засоби: драйвери, бібліотеки операцій і даних, менеджер бібліотек.

3.1 Серверний вузол системи та прикладні програми

Система автоматизованого проектування із застосуванням методу динамічної інтеграції даних являє собою структуру, що складається з Керуючого процесора (КП) і тематичних співпроцесорів: Графічного співпроцесора (ГП), табличний співпроцесора (ТБП), Математичного співпроцесора (МП) і Текстового співпроцесора (ТП). Робота забезпечується виконанням сценарію проектування Керуючим процесором. Сценарій проектування являє собою набір узагальнених операцій, що може бути відображений у вигляді графа. Узагальнена операція складається з декількох команд тематичних співпроцесорів, що мають загальну смислове завершеність.

Представлена програмна реалізація моделі середовища автоматизованого проектування найбільш повно відповідає рішенням завдань, покладених на систему для забезпечення динамічної інтеграції даних. У представлений реалізації виконується повне відпрацювання всіх узагальнених операцій, наявних в сценарії проектування. Створена структура, що відпрацьовує в'язку "команда - повідомлення" у всіх тематичних співпроцесорами. Реалізовано механізм передачі даних між процесорами.

Механізм передачі статичних даних, тобто декларативного представлення даних, представляє сукупність дій співпроцесорів, що оперують процедурним поданням даних. В результаті виконання послідовності команд, наявних в

узагальненій операції, процесори в певний момент часу забирають з Єдиного Інформаційного Простору (ЄІП) дані того типу, який були туди поміщені. Таким чином, сукупність операцій, виконаних над даними процедурного уявлення, виконує інтеграцію даних декларативного подання в єдиному інформаційному процесі.

Модель середовища автоматизованого проектування має структуру, представлену на рис.3.1.

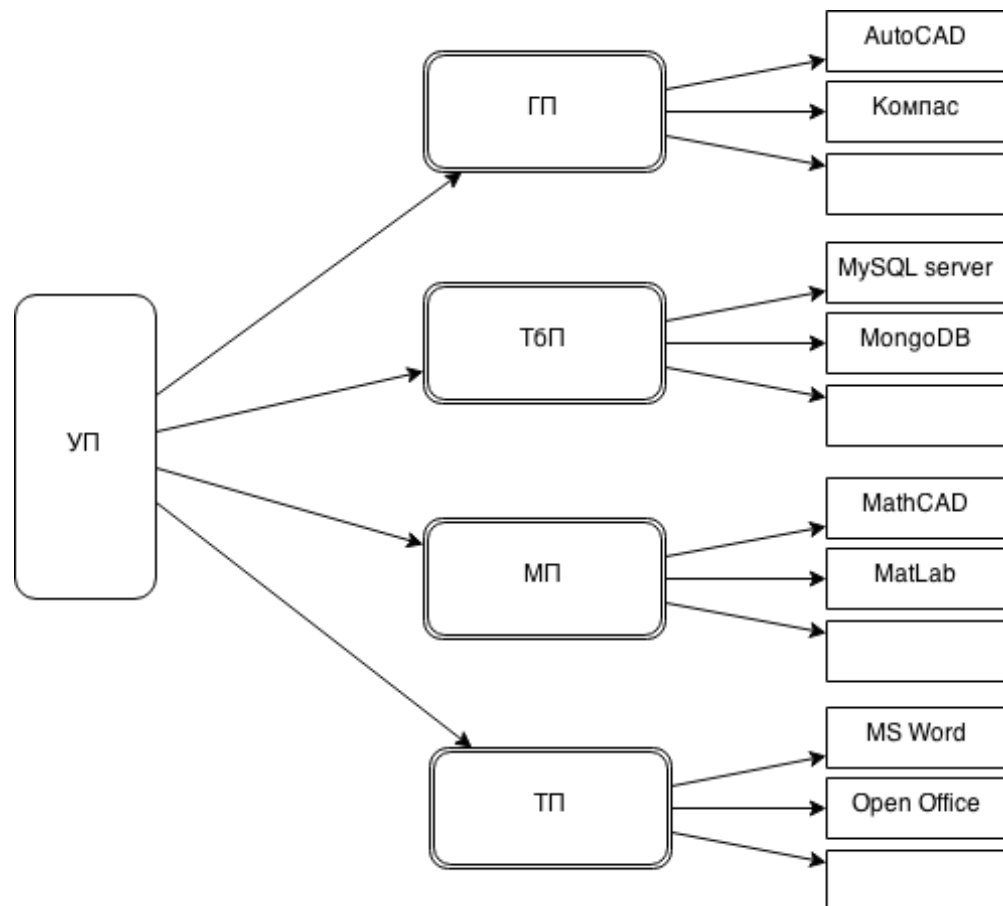


Рис. 3.1. Структурна схема інтегрованого середовища автоматизованого проектування

Керуючий процесор

Керуючий процесор підтримує виконання узагальнених операцій згідно зі сценарієм проектування. КП видає команди тематичним співпроцесорами і приймає від них тільки інформаційні повідомлення. Виконання кожної дії зі складу узагальненої операції починається з команди КП. Засобами зв'язку команда передається в тематичні співпроцесори, де вони обробляються відповідним чином. Після виконання дій команди тематичні співпроцесори

повертають управління в КП та через засоби зв'язку відсилають йому інформаційне повідомлення (рис. 3.2).

КП забезпечує роботу єдиного інформаційного простору:

- надає доступ кожному тематичному співпроцесору;
- регулює черговість доступу тематичних співпроцесорів до ЄП;
- обробляє дані, прийняті від одного тематичного співпроцесора в дані зрозумілі іншому тематичним співпроцесору.

КП також здійснює протоколювання дій вироблених проектувальником в процесі роботи.

При цьому КП:

Вивдає: команди тематичним співпроцесорами.

Приймає: повідомлення від тематичних співпроцесорів.

Керуючий процесор складається з наступних модулів:

- модуль управління;
- модуль ведення сценарію проектування;
- модуль ведення протоколу подій;
- модуль обробки повідомлень;
- модуль обробки команд;
- модуль підтримки єдиного інформаційного простору;
- модуль зв'язку

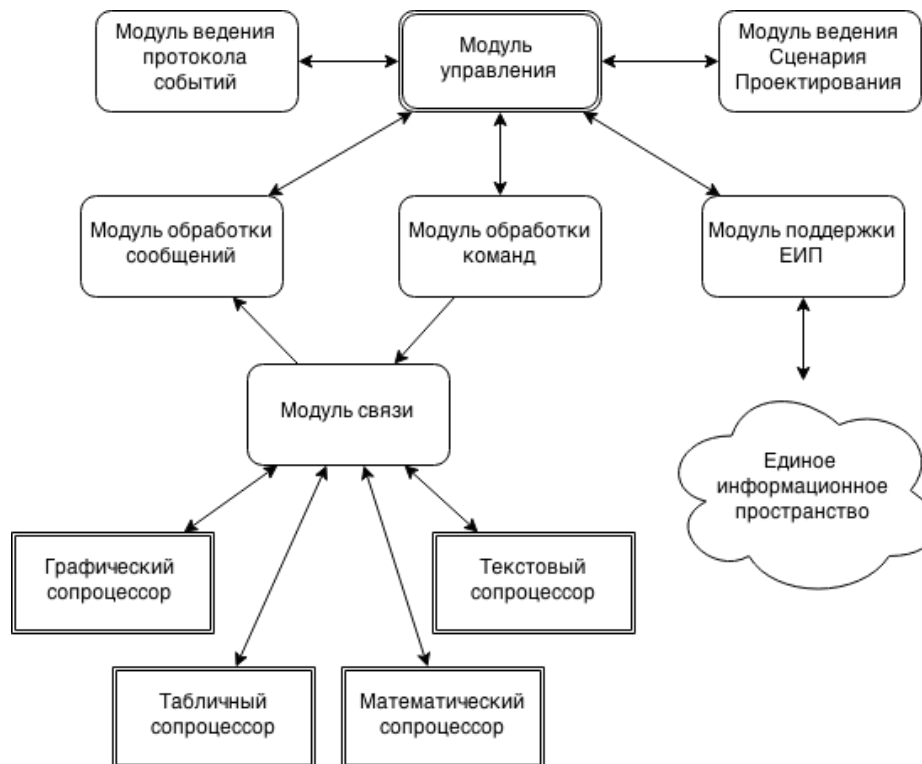


Рис. 3.2. Структурна схема Керуючого процесора

Модуль управління

Модуль управління регулює взаємовідносини модулів керуючого процесора між собою. Модуль виконує і підтримує такі функції:

- перерозподіл даних від одного модуля до іншого;
- опитування стану модулів;
- визначення пріоритету виконуваних завдань;
- передача керування модулями відповідно до визначеного пріоритетом.

Вивдає:

- модулю ведення сценарію проектування - повідомлення про завершення виконання попередньої команди з сценарію проектування;
- модулю ведення протоколу подій - команду про занесення команди у файл протоколу подій;
- модулю обробки повідомлень - запит на передачу повідомлення від тематичного співпроцесора;
- модулю обробки команд - текст команд з сценарію проектування;
- модулю підтримки єдиного інформаційного простору - необхідні службові дані для запису в єдиний інформаційний простір.

Приймає:

- від модуля ведення сценарію проектування - наступну команду за сценарієм проектування;
- від модуля ведення протоколу подій - повідомлення про виконання команди;
- від модуля обробки повідомлень - чергове прийняте повідомлення від тематичних співпроцесорів;
- від модуля обробки команд - повідомлення про відправку тексту команд модулю зв'язку;
- від модуля підтримки єдиного інформаційного простору - необхідні службові дані з єдиного інформаційного простору.

Модуль ведення сценарію проектування

Модуль ведення сценарію проектування відповідає за обробку файлу сценарію проектування. Модуль виконує і підтримує такі функції:

- перегляд і редагування файлу в текстовому вигляді;
- перегляд і редагування файлу у вигляді графа;
- зчитування команд з файлу і передачу в керуючий модуль;
- прийом повідомлень від модуля управління.

Видає:

- модулю управління - наступну команду за сценарієм проектування.

Приймає:

- від модуля управління - повідомлення про завершення виконання попередньої команди.

Модуль ведення протоколу подій

Модуль ведення протоколу подій відповідає за збереження всіх проведених дій у файлі протоколу подій. Модуль виконує і підтримує такі функції:

- перегляд файлу протоколу подій;
- обробка файлу протоколу подій;

Видає:

- модулю управління - повідомлення про виконання команди.

Приймає:

- від модуля управління - команду про занесення команди у файл протоколу подій.

Модуль обробки повідомлень

Модуль обробки повідомлень відповідає за прийом повідомлень від тематичних співпроцесорів через модуль зв'язку. Модуль виконує і підтримує такі функції:

- прийом тексту повідомлень від модуля зв'язку;
- накопичення повідомлень в буфері в порядку їх прибуття;
- видача повідомлень в модуль управління за запитом.

Видає:

- модулю управління - чергове прийняте повідомлення від тематичних співпроцесорів.

Приймає:

- від модуля управління - запит на передачу повідомлення від тематичного співпроцесора.
- від модуля зв'язку - текст інформаційного повідомлення від тематичних співпроцесорів.

Модуль обробки команд

Модуль обробки команд відповідає за передачу команд для тематичних співпроцесорів модулю зв'язку. Модуль виконує і підтримує такі функції:

- обробка команд для тематичних співпроцесорів з модуля ведення сценарію проектування;
- розбиття узагальнених операцій на складові команди і передача їх адресатам (тематичним співпроцесорами);
- передача тексту команд в модуль зв'язку.

Видає:

- модулю зв'язку - текст команд для тематичних співпроцесорів модулю зв'язку;
- модулю управління - повідомлення про відправку тексту команд модулю зв'язку.

Приймає:

- від модуля управління - текст команд з сценарію проектування.

Модуль підтримки єдиного інформаційного простору

Модуль забезпечує підтримку єдиного інформаційного простору і регулює доступ до нього тематичних співпроцесорів. Модуль виконує і підтримує такі функції:

- містить в пам'яті комп'ютера і на диску фізичний простір для занесення в нього даних;
- регулює доступ до простору тематичних співпроцесорів;
- забезпечує зберігання даних до їх запитання;
- видаляє непотрібні дані з ЄІП;
- зчитує і записує службові дані в ЄІП.

Видає:

- модулю управління - службові дані з ЄІП;
- єдиного інформаційного простору - службові дані для запису.

Приймає:

- від модуля управління - службові дані для запису в ЄІП;
- від ЄІП - службові дані для модуля управління.

Модуль зв'язку

Модуль зв'язку відповідає за підтримання зв'язку з тематичними співпроцесорами. Модуль виконує і підтримує такі функції:

- забезпечення зв'язку з тематичними співпроцесорами;
- прийом тексту команд, що посилаються в тематичні співпроцесори;
- відправка команд в тематичні співпроцесори;
- регулювання доступу тематичних співпроцесорів до керуючого процесору;
- прийом повідомлень від тематичних співпроцесорів і передача їх в модуль обробки повідомлень.

Видає:

- тематичним співпроцесорами - тексти команд;

- модулю обробки повідомлень - тексти отриманих інформаційних повідомлень.

Приймає:

- від модуля обробки команд - тексти команд для тематичних співпроцесорів;
- від тематичних співпроцесорів - тексти інформаційних повідомлень.

Графічний співпроцесор

Графічний співпроцесор є засобом обробки графічних даних. ГП забезпечує:

- зв'язок з графічними редакторами через драйвер Д,
- виконання команд КП,
- передачу інформаційних повідомлень в КП,
- видачу команд графічного редактору,
- отримання від графічного редактора інформаційних повідомлень.

ГП реалізує виконання графічних робіт процесу проектування та здійснює взаємодію графічних виконуючих процесорів з КП.

Табличний співпроцесор

Табличний співпроцесор являє собою засіб обробки табличних даних. ТбП забезпечує:

- зв'язок з СУБД через драйвер Д,
- виконання команд УП,
- передачу інформаційних повідомлень в КП,
- видачу команд СУБД,
- прийом від СУБД інформаційних повідомлень.

ТбП реалізує взаємодію процесу проектування з СУБД. Підтримує ведення внутрішніх службових баз даних.

Математичний співпроцесор

Математичний співпроцесор є засобом виконання математичних операцій. МП забезпечує:

- зв'язок з математичним редактором через драйвер Д;
- виконання команд КП;

- передачу інформаційних повідомлень в КП;
- видачу команд математичного редактору;
- отримання від математичного редактора інформаційних повідомлень.

МП реалізує виконання розрахункових операцій виникають у процесі проектування і підтримує взаємодію математичних виконуючих процесорів з КП.

Текстовий співпроцесор

Текстовий співпроцесор являє собою засіб обробки текстових даних. ТП забезпечує:

- зв'язок з текстовим редактором через драйвер Д;
- виконання команд КП;
- передачу інформаційних повідомлень в КП;
- видачу команд текстового редактору;
- отримання від текстового редактора інформаційних повідомлень.

ТП реалізує текстовий опис проєктованих об'єктів, використовуваного обладнання, вироблених обчислень і т.д. ТП підтримує взаємодію виконуючих процесорів обробки тексту з КП.

Таким чином, представлена вище структура серверного вузла середовища автоматизованого проектування демонструє можливість програмної реалізації реляційного методу динамічної інтеграції даних на рівні керуючого процесора. Представлений серверний вузол має можливість виконання основних функцій запропонованого керуючого процесора: ведення сценарію проектування, передача команд і прийом повідомлень від тематичних співпроцесорів, протоколювання дії проєктувальника.

3.2 Опис роботи модуля управління інтегрованої системи

3.2.1 Загальна схема роботи

Модуль управління відповідає за працездатність всієї системи в цілому. Відповідно, до нього висуваються самі різнобічні вимоги:

- Надання інструментарію користувачеві для роботи з системою;

- Робота з усіма компонентами системи (всі доступні тематичні співпроцесори);
- Робота з єдиним інформаційним простором і організація системи доступу до неї для решти компонентів системи.

Розглянемо основний принцип роботи модуля (рис. 3.3)

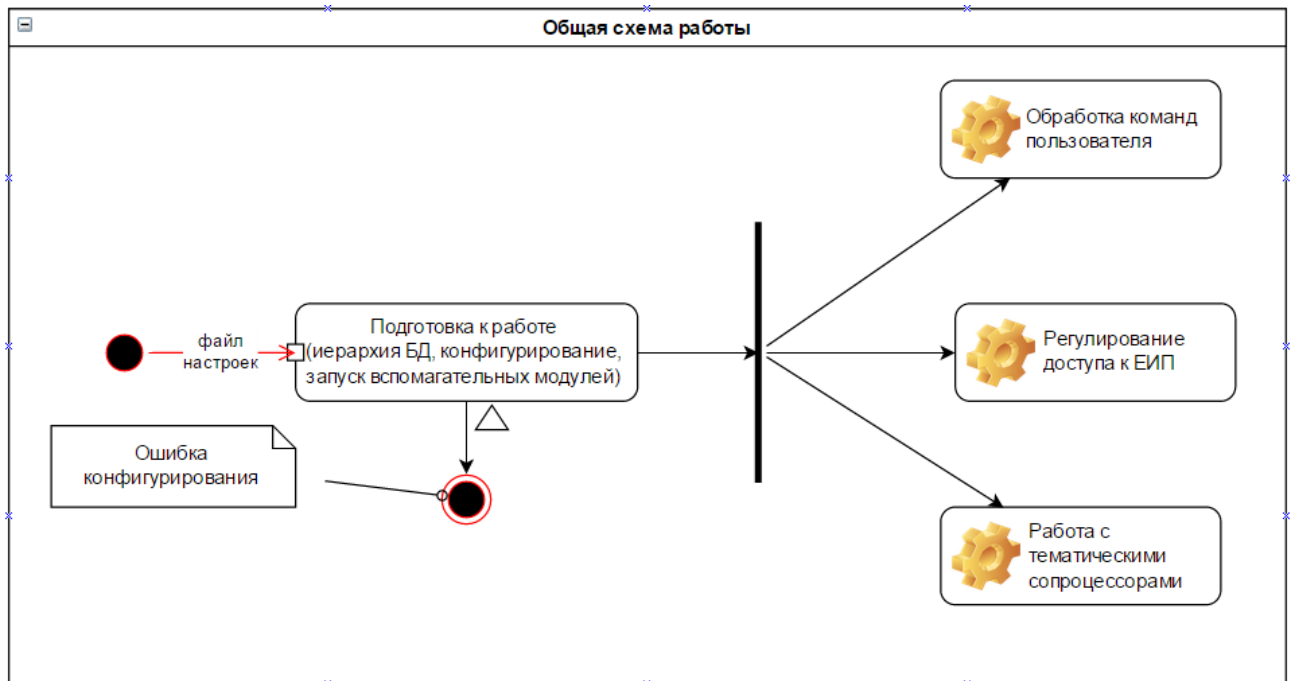


Рис. 3.3. Загальна схема роботи модуля управління

Модуль працює в два етапи:

1. Етап конфігурації системи;
2. Вічний робочий цикл.

Перший необхідний для налаштувань всіх компонентів системи. Для цього використовується конфігураційний файл, в якому зазначаються всі налаштування:

- Конфігурація єдиного інформаційного простору:
 - Параметри підключення;
 - Режими роботи;
 - Обмеження;
 - Політики безпеки;
- Специфікація типів даних, використовуваних для опису співпроцесорів;
- Конфігурація всіх доступних тематичних співпроцесорів;

- Назва співпроцесора;
- Опис;
- Список доступних функцій:
 - Назва;
 - Параметри підключення;
 - Опис функціоналу;
 - Опис вхідних параметрів;
 - Опис результату.

Так як всі компоненти взаємозамінні і спосіб реалізації кожного окремо взятого модуля і всієї системи в цілому багато в чому є справою смаку і навичок розробника ми змалюємо загальні принципи побудови і постараємося привести якомога більше наочних прикладів.

Розглянемо приклад такого конфігураційного файлу. В якості єдиного інформаційного простору виберемо СУБД MySQL. Мова конфігураційного файлу XML. Тоді блок конфігурування ЄП може виглядати наступним чином:

```
<?xml version="1.0" encoding="UTF-8"?>
<configurations>
  <database>
    <host>localhost</host>
    <port>3340</port>
    <db_name test</db_name>
    <user>some_user_name</user>
    <password>password</password>
    <option>CLIENT_SSL</option>
  </ database >
</configurations>
```

Далі приступаємо до безпосереднього опису системи. Для початку необхідно описати типи даних якими ми надалі будемо оперувати:

```
<types>
  <type name="point">
    <field>
      <name>x</name>
      <type>int</type>
    </field>
    <field>
      <name>y</name>
```

```

        <type>int</type>
    </field>
</type>
<type name="line">
    <field>
        <name>first_point</name>
        <type>point</type>
    </field>
    <field>
        <name>second_point</name>
        <type>point</type>
    </field>
</type>
<type name="circle">
    <field>
        <name>center</name>
        <type>point</type>
    </field>
    <field>
        <name>radius</name>
        <type>int</type>
    </field>
</type>
</types>

```

Тепер розглянемо приклад опису доступних співпроцесорів

```

<moduls>
    <module name="AutoCAD">
        <description>AutoCAD driver. Converts data from
internal format to AutoCAD format</description>
        <operation name="DrawCircle">
            <input type="point">center</input>
            <input type="int">radius</input>
            <output
type="circle">resulting_object</output>
        </operation>
        <operation name="DrawLine">
            <input type="point">first_point</input>
            <input type="point">second_point</input>
            <output
type="line">resulting_object</output>
        </operation>
    </module>
</moduls>

```

Після опису всіх конфігураційних параметрів ми маємо повний опис системи. В даному випадку були вказані параметри підключення до СУБД (ЄІП), до системи підключено один модуль (AutoCAD), що підтримує дві операції (DrawCircle і DrawLine).

Використовуючи подібний конфігураційний файл модуль управління має всю необхідну інформацію для організації роботи як з ЄІП так і з іншими модулями системи. Якщо інформація не повна або вказана некоректно система згенерує звіт про помилку і безпечно закінчить роботу.

Закінчивши всі підготовчі роботи, описані вище, модуль управління приступає до виконання своїх функцій, а саме:

- Обробка команд користувача - графічний інтерфейс, що дозволяє користувачеві працювати з системою;
- Регулювання доступу до ЄІП - маршрутизація запитів до ЄІП від решти компонентів системи, запобігання одночасного використання одного ресурсу в ЄІП;
- Робота з тематичними співпроцесорами - організація процесу обміну командами між усіма компонентами системи.

Кожна з поставлених завдань може виконуватися паралельно для забезпечення швидкодії всієї системи. Модуль управління переходить в режим постійного очікування команд в рамках будь-якої з поставлених вище завдань. Розглянемо докладніше механізми реалізації кожної з задач окремо.

3.2.2 Обробка команд користувача

Компонент відповідає за взаємодію системи і користувача за коштами графічного інтерфейсу. Компонент повинен забезпечити виконання всіх необхідних користувачеві завдань, причому виконання має бути інформативною (інформування користувача про процес виконання поточного завдання), необхідно передбачити можливість впливу користувача на процес проектування (переривання, паралельне виконання завдань, редагування параметрів проектування «на льоту»). Також користувач повинен мати у своєму розпорядженні прозорі механізми для конфігурації системи (див. пункт вище).

Для вирішення таких завдань пропонується наступна структурна схема компонента (рис 3.4).

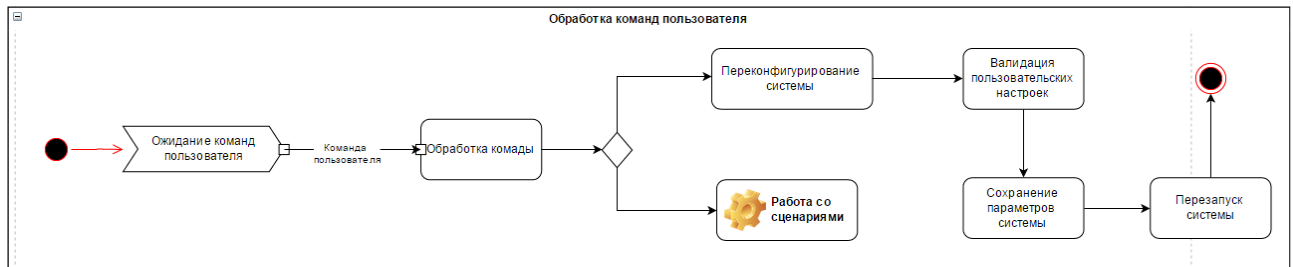


Рис. 3.4. Обработка команд користувача

Компонент весь час роботи системи знаходиться в очікуванні команд користувача. Виділяються два класи команд:

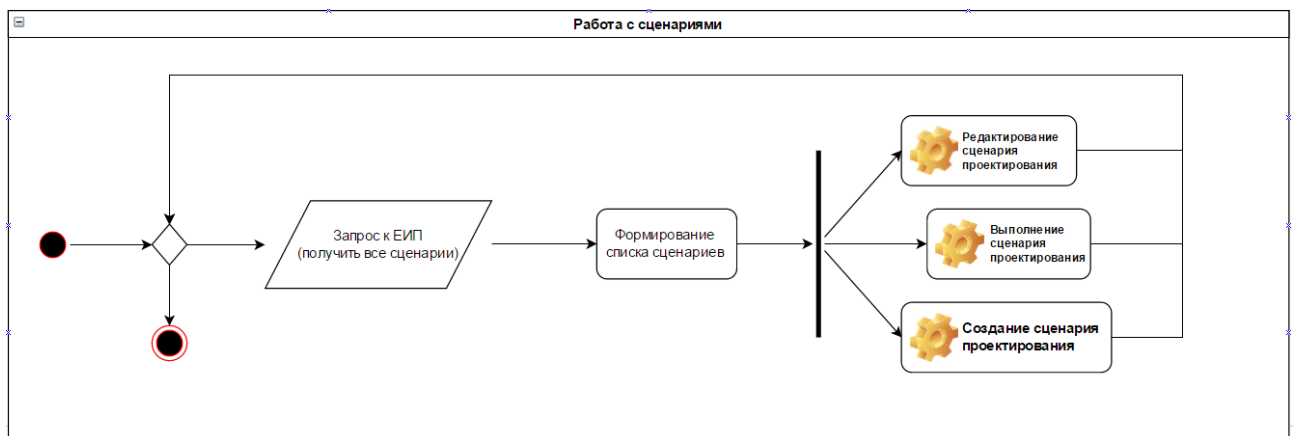
- Команди конфігурування;
- Команди проектування.

Команди конфігурування надають користувачеві можливість налаштування як системи в цілому так і її окремих компонентів. Результатом виконання таких команд є модифікований конфігураційний файл, описаний вище. Для виконання будь-якої команди даного класу необхідно виконати ряд обов'язкових дій:

1. Зробити резервну копію поточної конфігурації системи - у разі виникнення будь-яких помилок під час впровадження нових налаштувань завжди буде можливість повернутися до старої, робочої версії параметрів системи;
2. Перевірка нових налаштувань - всі зміни, внесені користувачем необхідно ретельно перевіряти щоб уникнути виникнення помилок користувача. Правила конфігурування системи повинні бути однозначними, достатньо суворими (користувачеві доступні тільки необхідні параметри, які доступні для зміни), для нормальної працездатності системи файл конфігурацій не повинен мати взаємовиключних параметрів;
3. Перетворення нових настрої до формату конфігураційного файлу, впровадження нових параметрів;
4. Так як багато параметрів системи налаштовуються тільки при запуску системи, для застосування нових параметрів її необхідно перезапустити.

Другий клас команд - це команди проектування. Це головні команди, що відповідають за доступність всього функціоналу системи для кінцевого користувача. Користувач управляє процесом проектування за допомогою сценарію проектування. Сценарій являє собою список інструкцій для модуля управління. Нижче розглянемо механізми роботи зі сценарієм і механізми його виконання.

3.2.3 Робота зі сценаріями



3.5. Работа зі сценаріями проектування

Для цих цілей розроблено компонент модуля управління. Основне його завдання - підтримка CRUD (створення, читання, зміна, видалення) операцій для роботи зі сценаріями. Всі вже існуючі сценарії зберігаються в ЄІП в заздалегідь обумовленому форматі (XML, JSON, YAML, та інші). Формат представлення даних не має особливого значення і залежить тільки від смаків розробника. Можна виділити лише основні рекомендації при виборі формату:

- Компактність даних - формат повинен додавати мінімум службової інформації до корисної інформації даними;
- Простота використання - інструментарій для роботи з форматом (серіалізація / десеріалізація) повинен бути зручний у використанні, швидким і надійним в роботі, так як обсяг оброблюваної інформації буде істотним;
- Читабельність - дані, збережені у вибраному форматі повинні бути зрозумілими і легко читаються людиною, або розробник повинен надати додатковий інструмент адаптації сценарію для користувача (рис. 3. 6).

Так як всі сценарії проектування (як готові так і ті, що потребують доопрацювання) зберігаються в ЄІП, насамперед необхідно їх прочитати. Це робиться відповідним запитом, формат якого залежить від реалізації самого ЄІП. На підставі відповіді формується список доступних скриптів. На цьому етапі необхідно показати користувачеві результат з коротким описом кожного сценарію і надати йому можливість вибрати для роботи будь-якої з них.

Тепер розглянемо реалізацію функціоналу для роботи зі сценаріями. На даному етапі виділяються три основні класи задач:

- Створення нового сценарію проектування;
- Редагування вже існуючого сценарію;
- Виконання сценарію проектування.

3.2.4 Створення нового сценарію проектування

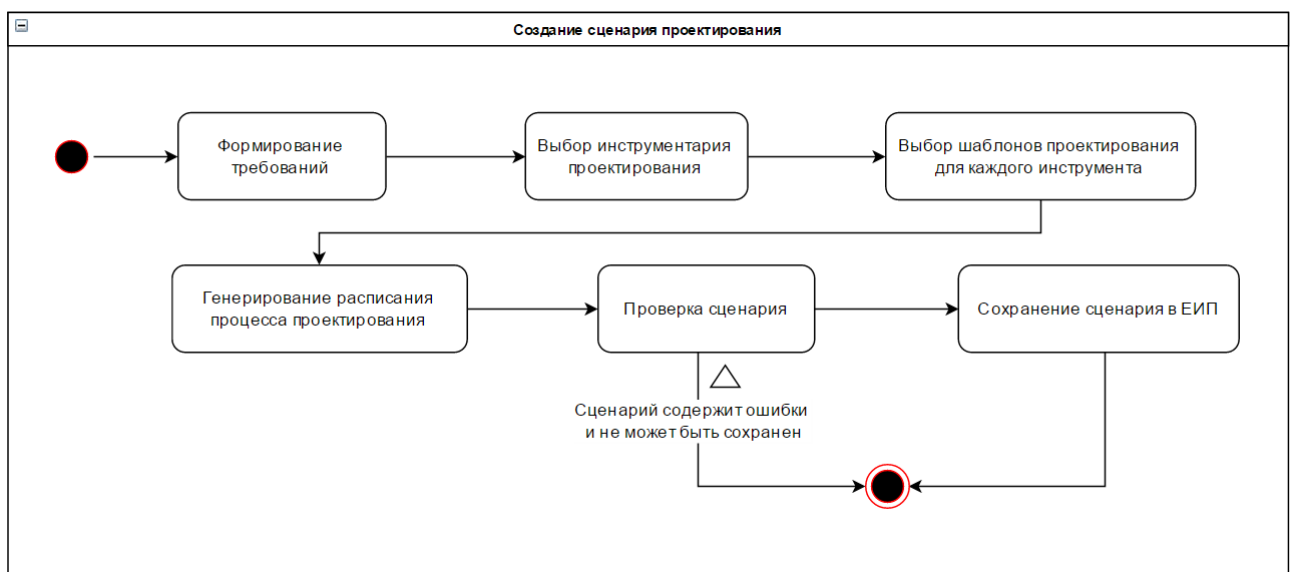


Рис. 3.6. Створення сценарію проектування

Передбачається, що створенням та редагуванням сценаріїв проектування буде займається тільки конструктор. Його завданням в даному випадку буде компоновка існуючих конструкцій в одне кінцеве готовий виріб. У зв'язку з цим створення нового сценарію пропонується розбити на наступні етапи:

1. Формування вимог - конструктор зазначає види робіт, необхідних для проектування виробу (проектування механічних компонентів,

проектування електричних схем, імітаційне моделювання, створення технічної документації);

2. Вибір інструментів проектування - для кожного зазначеного виду робіт конструктору пропонується на вибір кілька інструментів, серед яких конструктор вибирає оптимальне, на його погляд рішення;
3. Вибір шаблонів проектування для кожного інструменту - для кожного обраного інструменту конструктор ставить список завдань, необхідних до виконання. Система пропонує підлозі готові (шаблонізувати) рішення. Конструктор тільки обирає необхідне і вказує відсутні параметри (наприклад, ширину і довжину деталі). При цьому, після виконання всіх завдань маємо отримати готовий і робочий виріб;
4. Генерування розкладу процесу проектування - на даному етапі система має своєрідне технічне завдання, на підставі якого генерує на вибір конструктора два розкладу - оптимальне за часом роботи (з максимальним використанням обчислювальних потужностей комп'ютера) і оптимальне по ресурсах (менш інтенсивне використання обчислювальних ресурсів але більш витратне за часом виконання);
5. Перевірка сценарію - пошук можливих невідповідностей на всіх етапах проектування з метою уникнути некорінної роботи системи і мінімізувати помилки в готовому виробі після завершення всіх проєкційних робіт;
6. Збереження сценарію в ЄІП - перетворення готового сценарію проектування в формат ЄІП і його збереження.

3.2.5 Редагування сценарію проектування

Редагування сценарію проектування дозволяє вносити зміни у вже існуючі сценарієм з метою виправлення помилок конструювання, або з якоїсь іншої причини (рис. 3.7).

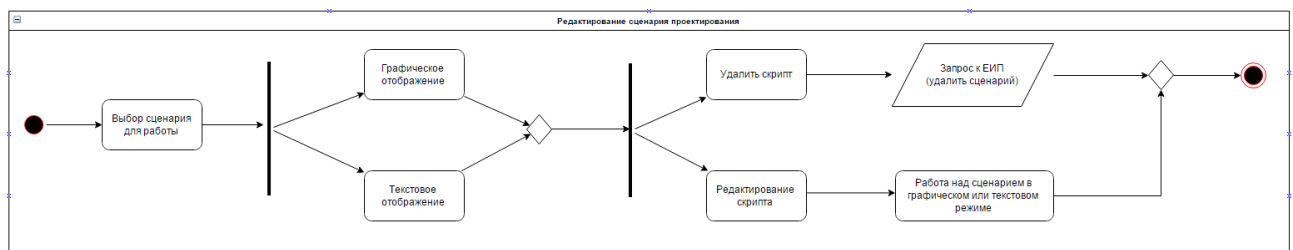


Рис. 3.7. Редагування сценарію проектування

Насамперед конструктор повинен вибрати сценарій, який він планує відредагувати. Далі система готує дві графічних форми подання сценарію:

- Текстовий - сценарій відображається в тому вигляді, в якому він зберігається в ЄП;
- Графічний - більш наочний спосіб представлення сценарію, який дозволяє конструктору вносити зміни не вникаючи в формат представлення сценарію в системі (наприклад, у формі блок-схеми або пазла). Конструктору демонструється обидві форми подання, з якими він може працювати одночасно, зміни в одній формі повинні на льоту підхоплюватися інший формою подання.
- Видалити скрипт - конструктор може видалити сценарій з системи (по суті з ЄП). Після даної операції сценарій більше не буде доступним для конструктора.
- Редагування скрипта - кожна форма подання сценарію надає свій набір інструментів для редагування файлу сценарію (зміна параметрів, додавання / видалення деяких функціональних блоків).

3.2.6 Виконання сценарію проектування

Компонент відповідає виконання окремої сценарію проектування, створеного конструкторами на попередніх етапах. У його обов'язки також входить синхронізація всіх дій системи в рамках виконання процесу проектування. Після виконання сценарію компонент надає конструктору звіт про виконану роботу а також, власне, результат проектування - готову проектну документацію по виробу (рис. 3.8).

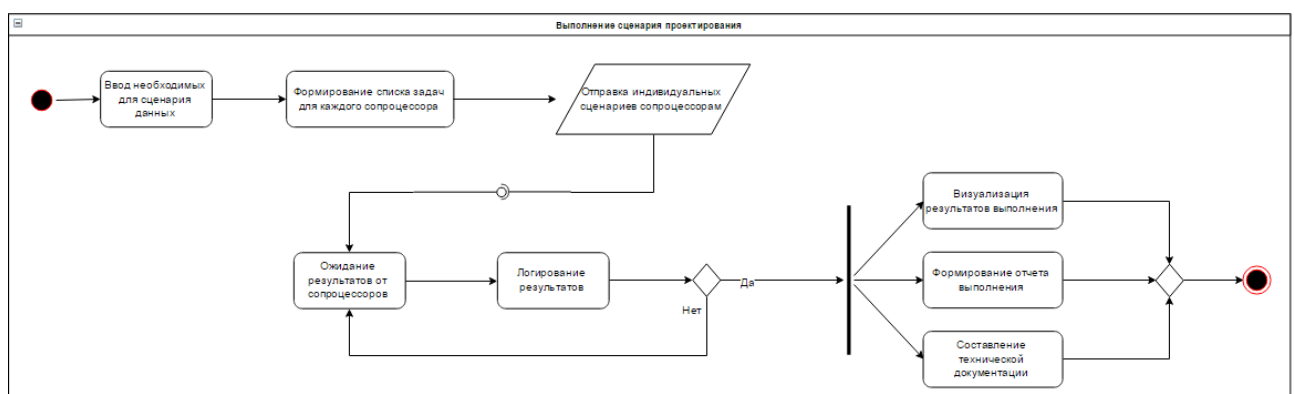


Рис. 3.8. Виконання сценарію проектування

Алгоритм роботи компонента:

1. Введення необхідних для сценарію даних - так як сценарій являє собою шаблон процесу проектування, йому необхідна параметризація бажаних параметрів. На даному етапі конструктору необхідно ввести всі ети параметри (наприклад, геометричні параметри виробу, електротехнічні характеристики, і т.д.). Далі система виконує перевірку вхідних даних для запобігання можливих помилок при подальшому проектуванні виробу;
2. Формування списку завдань для кожного тематичного співпроцесора - після деталізації шаблону проектування система приступає до формування повного списку завдань для конкретного сценарію. Всі завдання необхідно розділити між співпроцесорами. Таким чином в кінці отримуємо розклад зі списком завдань для кожного співпроцесора. Наступним важливим етапом є складання розкладу робіт з проектування. Система повинна гарантувати, що кожен співпроцесор при виконанні окремо взятої задачі матиме в своєму розпорядженні всі необхідні вхідні дані. Крім того розклад повинно бути оптимізовано за часом;
3. Відправлення індивідуальних сценаріїв співпроцесорами - кожен співпроцесор отримує свій окремий список завдань, складених на попередньому кроці. Надалі кожен співпроцесор незалежно один від одного може працювати над своїм сценарієм. Хід виконання контролюється модулем управління за допомогою команд, в яких співпроцесор отримує команду для виконання наступного завдання та відсутні вхідні дані для неї;
4. Цикл проектування - кожен співпроцесор приступає до виконання свого списку завдань, попередньо отримавши необхідні дані від модуля управління. Після виконання кожного завдання співпроцесор відправляє результат модулю управління і чекає команди для продовження роботи. Так продовжується до тих пір поки не буде виконано весь сценарій проектування. Всі дії на даному етапі

логуються модулем управління для формування звіту про процес проектування;

5. Формування звіту - після виконання сценарію проектування необхідно надати конструктору звіт про виконану роботу разом з результатами:

- а. Візуалізація результатів виконання - готові креслення виробу;
- б. Формування звіту виконання - звіт по кожній виконане завдання (використовуються логи, отримані в процесі виконання завдань);
- с. Складання технічної документації - автоматична генерація технічної документації на основі виконаної роботи і сценарію проектування.

3.2.7 Робота з тематичними співпроцесорами

Компонент відповідає за взаємодію модуля управління з усіма тематичними співпроцесорами, підключеними до системи, що в свою чергу дозволяє організувати взаємопов'язану роботу всіх співпроцесорів. Компонент надає однаковий для всіх співпроцесорів формат обміну повідомленнями та однаковий набір команд, які адаптуються кожним співпроцесором зокрема (рис. 3.9).

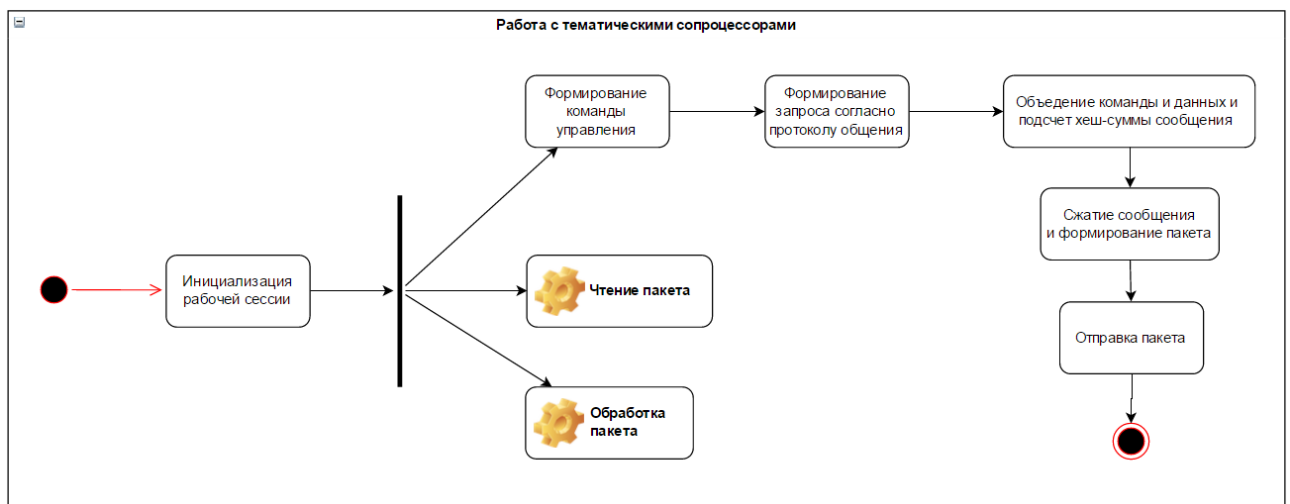


Рис. 3.9. Работа с тематичними співпроцесорами

Для роботи з усіма співпроцесорами компонент повинен виконувати наступні завдання:

- Ініціалізація робочої сесії - тут компонент встановлює з'єднання з конкретним співпроцесором і організовує стабільний канал для передачі даних;
- Читання пакету - читання пакета від співпроцесора, перевірка на коректність та валідація відправника;
- Обробка пакета - реакція модуля управління на отримані дані (див. нижче);
- Відправлення пакета - відправка повідомлення співпроцесору.

Алгоритм відправки керуючого повідомлення співпроцесору:

1. Формування команди управління - формування набору даних, які необхідно відправити співпроцесору;
2. Формування запиту згідно протоколу спілкування - структурування даних згідно шаблону повідомлення;
3. Об'єднання команди і даних, підрахунок хеш суми сполучення - підрахунок хеш суми для перевірки валідності пакету сопроцессором;
4. Стиснення повідомлення і формування пакета - для зменшення обсягу переданих даних отримане повідомлення стискається, таким чином отримуємо готовий до відправки пакет;
5. Відправлення пакету.

3.2.8 Читання пакету

Компонент відповідає за перетворення сирих даних, отриманих від інших компонентів системи в готові для обробки команди з даними (рис. 3.10).

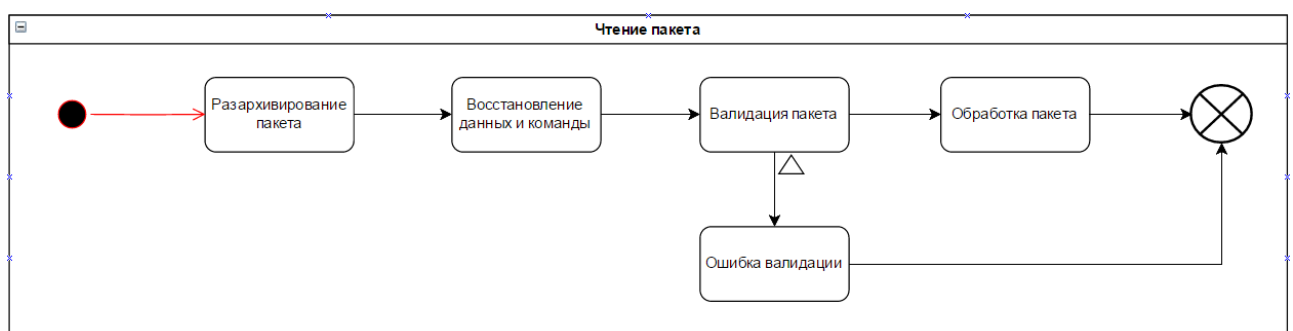


Рис. 3.10. Читання пакету

Алгоритм роботи компонента:

1. Розархівування пакета - так як для зменшення пакета при передачі він був архівувати, насамперед необхідно його відновити;
2. Відновлення даних і команди - з отриманих даних формуємо вихідне повідомлення;
3. Валідація пакета - підрахуємо контрольну хеш суму повідомлення і перевіряємо її з надісланої разом з пакетом. Якщо хеш суми не збігаються, пакет далі не обробляється;
4. Обробка пакета - відповідно до отриманої командою викликається обробник (див. нижче).

3.2.9 Обробка пакета

Компонент відповідає за реакцію системи на одержувані повідомлення. Для компонента важливо залишатися в робочому стані не дивлячись на дані, які він отримує. Тобто повинна бути гарантована висока відмовостійкість компонента (рис. 3.11).

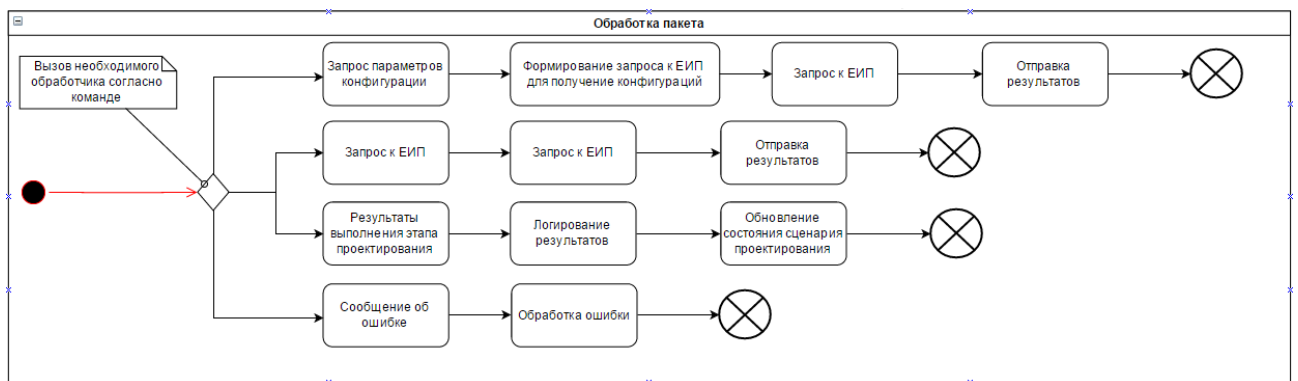


Рис. 3.11. Обробка пакета

У компоненті виділено 4 різних обробники команд і, відповідно, різні реакції системи на повідомлення. Розглянемо їх докладніше:

1. Запит параметрів конфігурації:
 - а. Формування запиту до ЄІП для отримання конфігурації системи
 - б. Виконання запиту
 - с. Відправка результату
2. Запит до ЄІП:

- а. Запит до ЄІП
 - б. Відправка результату
- 3. Результат виконання етапу проектування
 - а. Логування результат
 - б. Оновлення стану сценарію проектування
- 4. Повідомлення про помилку
 - а. Обробка помилок

3.2.10 Регулювання доступу до ЄІП

Єдиний інформаційний простір являє собою сховище всіх необхідних системі даних, відповідно, багато модулів системи мають незалежний один від одного доступ до нього. Тому необхідний компонент який буде регулювати доступ до ЄІП і запобігати можливим конфліктам (одночасне читання даних одним модулем і їх оновлення іншим, одночасне оновлення одних і тих же даних кількома модулями, і т.д.) (рис. 3.12).



Рис. 3.12. Регулювання доступу до ЄІП

Компонент постійно знаходиться в режимі очікування запитів до ЄІП. Тільки він має право на пряму працювати з ЄІП. Всі інші модулі звертаються до ЄІП через компонент.

Алгоритм виконання запиту до ЄІП:

1. Перевірка доступу до ресурсу, необхідного для виконання запиту - якщо необхідний ресурс зайнятий, запит додається в чергу і буде виконаний після всіх запитів в черзі. Якщо необхідний ресурс доступний, переходимо до наступного етапу;

2. Блокування ресурсу - блокуємо ресурс, щоб наступні запити не змогли змінити його в процесі виконання запиту;
3. Виконання запиту - власне, виконання самого запиту. Тут все залежить від інструменту, що використовується в якості ЄП. Якщо це реляційна БД, то виконується SQL запит, якщо просто набір файлів, то виконується пошук по необхідним файлам;
4. Звільнення ресурсу - тепер займаний ресурс можна звільнити і інші запити можуть його використовувати;
5. Відправлення результату - відправка клієнту отриманого в результаті запиту результату.

3.3 Засоби реалізації динамічної інтеграції даних

Тематичні співпроцесори

Тематичний співпроцесор є засобом обробки даних власного формату. Сам тематичний співпроцесор не виробляє дій безпосередньо змінюють даних в файлах проекту і на екрані. Тематичний співпроцесор тільки передає певні команди відповідним Виконуючим процесорам (ВП). Виконуючі процесори підключаються до тематичного співпроцесору через відповідні драйвери. Тематичний співпроцесор здійснює трансляцію даних із загального процесорного мови в мову виконуючого процесора. Через засоби зв'язку тематичний співпроцесор передає команди виконуючому процесору і отримує від нього тільки інформаційні повідомлення (рис. 3.13).

Видає:

- керуючому процесору - інформаційні повідомлення про виконання команди;
- виконуючим процесорам - команди для виконання.

Приймає:

- від керуючого процесора - команди для виконання виконуючими процесорами;
- від виконуючих процесорів - інформаційні повідомлення.

Тематичні співпроцесори складаються з наступних модулів:

- модуль управління;
- модуль зв'язку з КП;
- модуль обробки вихідних повідомлень;
- модуль обробки вхідних команд;
- модуль зв'язку з ВП;
- модуль обробки вихідних команд;
- модуль обробки вхідних повідомлень;
- модуль підтримки єдиного інформаційного простору;
- єдиний інформаційний простір;
- драйвери.

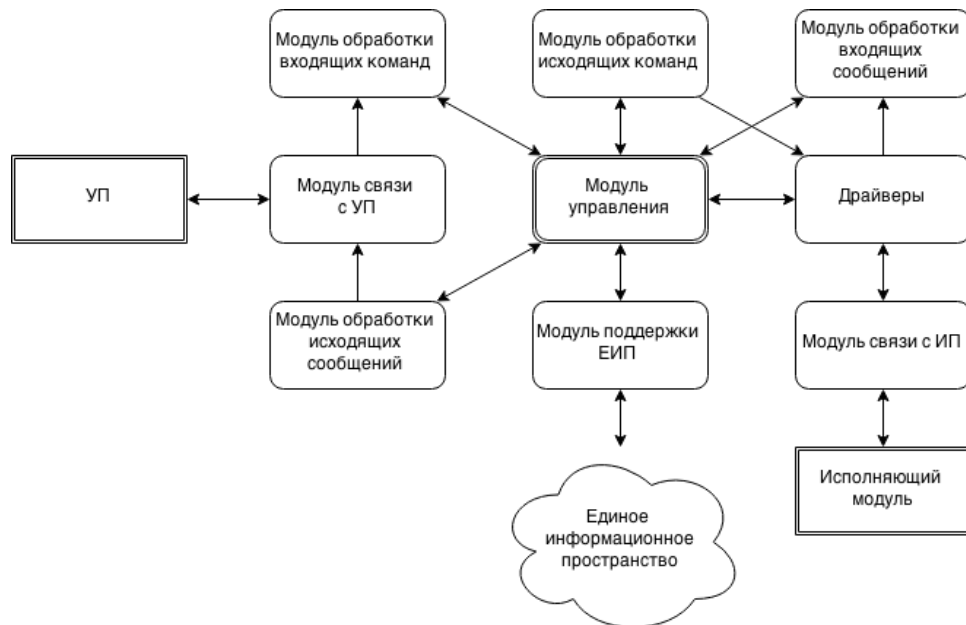


Рис. 3.13. Структурна схема тематичного співпроцесора

Модуль управління

Модуль управління регулює взаємовідносини модулів тематичного співпроцесора між собою. Модуль виконує і підтримує такі функції:

- перерозподіл даних від одного модуля до іншого;
- опитування стану модулів;
- визначення пріоритету виконуваних завдань;
- передача керування модулями відповідно до визначеного пріоритетом.

Видає:

- модулю обробки вихідних повідомлень - запит на вьщачу команди від керуючого процесора;
- модулю обробки вхідних команд - запит на видачу команди від керуючого процесора;
- модулю обробки вихідних команд - команду створення на створення команди для виконуючого процесора;
- модулю обробки вхідних повідомлень - запит на передачу інформаційного повідомлення;
- модулю підтримки єдиного інформаційного простору - дані для запису в єдиний інформаційний простір;
- драйверам - службову інформацію про характер роботи виконуючого процесора.

Приймає:

- від модуля обробки вихідних повідомлень - повідомлення про виконання роботи;
- від модуля обробки вхідних команд - команду від керуючого процесора;
- від модуля обробки вихідних команд - інформаційне повідомлення про виконання роботи;
- від модуля обробки вхідних повідомлень - оброблене інформаційне повідомлення;
- від модуля підтримки єдиного інформаційного простору - дані з єдиного інформаційного простору;
- від драйверів - службову інформацію про характер роботи виконуючого процесора.

Модуль зв'язку з УП

Модуль зв'язку з УП відповідає за підтримання зв'язку з керуючим процесором.

Модуль виконує і підтримує такі функції:

- забезпечення зв'язку з КП;
- прийом тексту інформаційних повідомлень посилаються в КП;

- відправка інформаційного повідомлення в КП;
- прийом команд від КП і передача їх в модуль обробки вхідних команд;
- накопичення інформаційних повідомлень в буфері по мірі їх прибуття.

Вивдає:

- керуючому процесору - інформаційні повідомлення про виконання команд тематичним співпроцесором;
- модулю обробки вхідних команд - команди, одержувані від КП.

Приймає:

- від керуючого процесора - команди для виконання тематичним співпроцесором;
- від модуля обробки вихідних повідомлень - текст інформаційних повідомлень про виконання команд тематичним співпроцесором.

Модуль обробки вихідних повідомлень

Модуль обробки вихідних повідомлень відповідає формування інформаційних повідомлень для керуючого процесора. Модуль виконує і підтримує такі функції:

- формує текст інформаційних повідомлень;
- передає сформований текст в модуль зв'язку з КП.

Вивдає:

- модулю зв'язку з КП - текст інформаційних повідомлень для КП;
- модулю управління - повідомлення про виконання роботи.

Приймає:

- від модуля управління - команду для створення тексту інформаційного повідомлення.

Модуль обробки вхідних команд

Модуль відповідає за прийом від керуючого процесора вхідних команд для виконання тематичними співпроцесорами. Модуль підтримує і виконує такі функції:

- прийом тексту команд від модуля зв'язку;

- накопичення повідомлень в буфері в порядку їх прибуття; перетворення тексту команди в команду для модуля управління; видача команд в модуль управління за запитом.

Видає:

- модулю управління - команду від КП.

Приймає:

- від модуля зв'язку з КП - текст команди КП;
- від модуля управління - запит на видачу команди від КП.

Модуль зв'язку з ВП

Модуль зв'язку з ВП забезпечує зв'язок з виконуючими процесорами.

Модуль підтримує і виконує такі функції:

- забезпечення зв'язку з виконуючим процесором;
- прийом тексту інформаційних повідомлень від виконуючих процесорів;
- передача тексту інформаційних повідомлень в модуль обробки вхідних повідомлень;
- прийом команд від модуля обробки вихідних команд і передача їх у виконуючий модуль.

Видає:

- виконуючому модулю - текст команд для виконання;
- драйверу - текст інформаційних повідомлень прийнятих від виконуючого модуля.

Приймає:

- від виконуючого модуля - текст інформаційних повідомлень;
- від драйвера - текст команд для передачі в виконуючий процесор.

Модуль обробки вихідних команд

Модуль обробки вихідних команд відповідає за передачу команд для виконуючих процесорів драйверу. Модуль виконує і підтримує такі функції:

- обробка команд для виконуючих процесорів від модуля управління, передача тексту команд в драйвер.

Видає:

- модулю управління - інформаційне повідомлення про виконання роботи;

- драйверу - текст команд для виконання виконуючими процесорами.

Приймає:

- від модуля управління - команду на створення команди для виконуючого модуля.

Модуль обробки вхідних повідомлень

Модуль обробки вхідних повідомлень відповідає обробку вхідних інформаційних повідомлень від виконуючого модуля. Модуль підтримує і виконує такі функції:

- приймає текст інформаційних повідомлень від виконуючих процесорів;
- накопичує інформаційні повідомлення в буфері по мірі їх надходження;
- обробляє і передає інформаційне повідомлення модулю управління за запитом.

Видає:

- модулю управління - оброблене інформаційне повідомлення.

Приймає:

- від модуля управління - запит на передачу інформаційного повідомлення;
- від драйвера - текст інформаційного повідомлення.

Модуль підтримки єдиного інформаційного простору тематичного співпроцесора

Модуль забезпечує доступ тематичних співпроцесорів в єдиний інформаційний простір. Модуль виконує і підтримує такі функції:

- регулює доступ до єдиного інформаційного простору виконуючих процесорів;
- забезпечує прийом і передачу даних для тематичного співпроцесора з єдиного інформаційного простору;
- зчитує і записує службові дані в простір.

Видає:

- керуючому процесору - дані з єдиного інформаційного простору;
- єдиного інформаційного простору - дані для запису.

Приймає:

- від керуючого процесора - дані для запису в єдиний інформаційний простір;
- від єдиного інформаційного простору - дані для модуля управління.

Драйвери

Драйвер призначений для трансляції команд тематичних співпроцесорів в команди виконуючого процесора. Драйвер призначений тільки для виконання технічних службових завдань та впливу на процес проектування не надає. Все управління драйвером проводиться тільки відповідним тематичним співпроцесором.

Драйвер створюється під конкретний виконуючий процесор і враховує всі його специфічні особливості, здійснюючи приведення вбудованої мови виконуючого процесора до процесорної мови тематичного співпроцесора.

Драйвер підтримує і виконує такі функції:

- приймає від модуля обробки вихідних команд текст команд переданої виконуючим процесорам;
- транслює текст команд в формат зрозумілий конкретному виконуючому процесору і передає його в модуль зв'язку третього рівня;
- приймає від модуля зв'язку з ІІ текст інформаційного повідомлення виконуючого процесора;
- транслює текст інформаційних повідомлень у формат тематичного співпроцесора і передає його в модуль обробки вхідних повідомлень.

Видає:

- модулю управління - службову інформацію про характер роботи виконуючого процесора;
- модулю обробки вхідних повідомлень - текст інформаційних повідомлень від виконуючих процесорів;
- модулю зв'язку з ВІІ - текст команд виконуючому процесору.

Приймає:

- від модуля управління - службові команди для драйвера;
- від модуля обробки вихідних повідомлень - текст команд переданої виконуючим процесорам;

- від модуля зв'язку третього рівня - текст інформаційного повідомлення виконуючого процесора.

Виконуючий процесор

Виконуючий процесор являє собою програмний продукт, що забезпечує виконання певних дій над об'єктом проектування. Виконуючі процесори поділяються на кілька груп за характером взаємодії з тематичними співпроцесорами. Кожен виконуючий процесор повинен відповідати наступним умовам:

- мати можливість зв'язку з іншими програмними засобами;
- мати засоби внутрішнього мови програмування або внутрішнього командного мови.

Виконуючий процесор підключається до тематичного співпроцесору через відповідний драйвер, який здійснює суміщення команд виконуючого процесора і тематичного співпроцесора (рис. 3.14).

Виконуючий процесор складається і наступних модулів:

- модуль управління;
- модуль зв'язку;
- модуль підтримки єдиного інформаційного простору;
- засіб виконання.

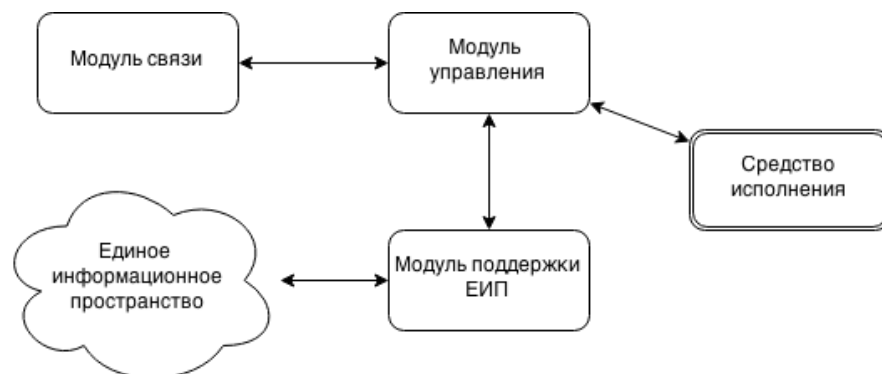


Рис. 3.14. Структурна схема виконуючого процесора

Модуль управління

Модуль управління регулює взаємовідносини модулів виконуючого процесора між собою. Модуль виконує і підтримує такі функції:

- перерозподіл даних від одного модуля до іншого;
- опитування стану модулів;

- визначення пріоритету виконуваних завдань;
- передача керування модулями відповідно до визначеного пріоритетом.

Модуль видає:

- засобу виконання - команди на виконання;
- модулю зв'язку - тексти інформаційних повідомлень;
- модулю підтримки єдиного інформаційного простору - дані для запису в єдиний інформаційний простір.

Модуль приймає:

- від засобу виконання - інформаційні повідомлення;
- від модулю зв'язку - отримані команди;
- від модулю підтримки єдиного інформаційного простору - дані з єдиного інформаційного простору.

Модуль зв'язку

Модуль зв'язку відповідає за підтримання зв'язку з тематичними співпроцесорами. Модуль виконує і підтримує такі функції:

- забезпечення зв'язку з тематичними співпроцесорами;
- прийом команд, що посилаються тематичними співпроцесорами;
- передача повідомлень в тематичні співпроцесори.

Видає:

- тематичним співпроцесорами - тексти інформаційних повідомлень;
- модулю управління - отримані команди.

Приймає:

- від тематичних співпроцесорів - тексти команд;
- від модулю управління - тексти інформаційних повідомлень.

Модуль підтримки єдиного інформаційного простору виконуючого процесора

Модуль забезпечує доступ виконуючих процесорів в єдиний інформаційний простір. Модуль виконує і підтримує такі функції:

- забезпечує прийом і передачу даних для виконавчих процесорів з єдиного інформаційного простору;

- зчитує і записує дані в єдиний інформаційний простір.

Вивдає:

- модулю управління - дані з єдиного інформаційного простору;
- єдиного інформаційного простору - дані для запису.

Приймає:

- від модуля управління - дані для запису в єдиний інформаційний простір;
- від єдиного інформаційного простору - дані для модуля управління.

Засіб виконання

Засіб виконання являє собою програмний продукт, що забезпечує виконання команд тематичного співпроцесора і команд проектувальника. Засобом виконання є програмний продукт, що має можливість автономно виконувати коло завдань необхідний у процесі проектування. Програмні продукти поділяються за належністю до тематичних співпроцесорами:

- Графічний процесор — AutoCAD, SolidWorks,...
- Табличний процесор — Access, dBase,...
- Текстовий процесор — MS Word, PageMaker,...
- Математичний процесор — MathCAD, Mathematica,...

Вивдає:

- модулю управління - інформаційні повідомлення.

Приймає:

- від модуля управління - команди на виконання.

Єдиний інформаційний простір

Єдиний інформаційний простір являє собою чергу структурованих записів вказівного типу. Процесори поміщають в дані у верхню частину черги, там відбувається їх накопичення, і в міру необхідності процесори забирають дані з нижньої частини черги. Таким чином, із застосуванням механізму черзі реалізується завдання отримання процесорами саме тих даних, які потрібні процесорам в даний момент.

Для вирішення завдання зберігання даних будь-яких типів в єдиному інформаційному просторі використовується механізм обробки вказівного типу. Мінлива вказівного типу містить у собі тільки адресу комірки пам'яті, в якій

зберігається запис. Шляхом конвертації даних з вказівного типу в необхідний і назад процесори отримують доступ до того типу даних, який їм потрібно в даний момент. За відповідність типів положеннях і вилучених даних відповідає сценарій проектування.

Оскільки в процесі обміну даними між процесорами виникає необхідність передавати кілька значень одночасно, то доцільно застосувати структурований запис. Для того, щоб передати кілька даних через одну змінну в єдиному інформаційному просторі, необхідно виконати дві команди КП і отримати інформаційні повідомлення від передавального і приймаючого процесора. Таким чином, при використанні однієї змінної використовується чотири такти процесора. При передачі n значень кількість тактів процесора, необхідних на передачу даних зростає відповідно в $n-1$ раз. Тому доцільно використовувати структурований запис з декількох змінних, де кількість змінних більше або дорівнює n .

Таким чином, представлена вище структура тематичних співпроцесорів і виконуючих процесорів середовища автоматизованого проектування демонструє можливість програмної реалізації реляційного методу динамічної інтеграції даних. Представлені програмні компоненти мають можливість виконання основних функцій запропонованих тематичних співпроцесорів: виконання узагальнених операцій з сценарію проектування, прийом команд і передача повідомлень від керуючого процесора, протоколювання дії проектувальника.

3.4 Бібліотеки узагальнених операцій і даних. Драйвери.

Бібліотеки узагальнених операцій і даних

Для більш ефективного вирішення конкретних завдань проектування потрібно застосовувати додаткові спеціальні методи. Для застосування спеціальних методів потрібні набори бібліотек даних, що полегшують рішення спеціалізованих завдань. Застосування доповнень до системи автоматизованого проектування призводить до значної економії часу проектувальника і дозволяє

йому ефективніше вирішувати поставлену задачу, не заглиблюючись у вивчення супутніх галузей науки, в яких описуються застосовувані ним методи.

Із застосуванням доповнень до системи автоматизованого проектування вирішуються такі завдання як:

- вибір необхідного обладнання з електронних довідників;
- виконання складних математичних розрахунків;
- детальний аналіз проектного устаткування;
- розрахунок найбільш ефективного розміщення обладнання на кресленні, найбільш економічного його підключення і т.д.

Пропонована середу автоматизованого проектування дозволяє підключати до себе необмежену кількість бібліотек узагальнених операцій і даних. Зручна форма інтеграції підключених даних в єдиному інформаційному процесі дозволяє в будь-який момент звернутися до бібліотек. Використання бібліотек можливе як по прямій команді проектувальника, так і за запитом виконуваних в даний момент узагальнених операцій. Подані бібліотеки даних дозволяють групувати повноцінні доповнення до проектної документації. Це можуть бути специфікації, таблиці характеристик обладнання тощо.

Додаткові бібліотеки в основному групуються по відношенню до вирішення конкретних завдань проектування. Вміст використовуваних бібліотек визначається проектувальником. Бібліотеки складаються як з довідкових даних, так і з наборів узагальнених операцій. Використовуючи менеджер бібліотек (див. Нижче), проектувальник може вибрати тільки бібліотеки узагальнених операцій або тільки бібліотеки даних і згрупувати з них загальний набір бібліотек, найбільш повно відповідає вимогам завдання.

Завдяки гнучкості компонентів, що входять до складу застосовуваних бібліотек і зручного менеджера бібліотек, що дозволяє групувати компоненти бібліотек відповідно до вимог розв'язуваної задачі, бібліотеки являють собою зручне і ефективний засіб для полегшення вирішення поставленої нестандартної задачі.

Бібліотека являє собою набір файлів, які зберігають в собі всі необхідні узагальнені операції і дані для проектування. Бібліотеки діляться на два типи:

бібліотеки узагальнених операцій і бібліотеки даних. Кількість підключаються бібліотек кожного типу не обмежена. Бібліотеки узагальнених операцій відносяться до керуючого процесору, а бібліотеки даних використовуються тематичними співпроцесорами. Кожен тип бібліотек відноситься до певної галузі проектування. За узгодженістю бібліотек зобов'язаний стежити проектувальник.

У бібліотеках узагальнених операцій знаходяться файли, що містять в собі набори команд для керуючого процесора. У бібліотеках даних зберігаються файли, призначені для використання тематичними співпроцесорами. Для ГП файли містять набір «примітивів» і «прототипів». Для ТБП - реляційну базу даних. Для МП - набір формул і констант, що застосовуються для математичних обчислень. Для ТП - текст, використовуваний в текстовому опис проекту. Кожна бібліотека містить в собі файл, в якому реєструються всі компоненти, що входять в бібліотеку.

Кількість узагальнених операцій в бібліотеці визначається упорядником бібліотеки, виходячи з вимог, висунутих бібліотеці. Узагальнені операції безпосередньо не пов'язані з даними, що зберігаються в бібліотеці даних, а мають тільки відкриту посилання, прив'язувати до конкретних даних у процесі проектування.

Дані, що входять до складу бібліотеки, являють собою набори статичних даних, що використовуються в процесі проектування. Дані поділяються за типами співпроцесорів. Кожен тематичний співпроцесор в системі оперує тільки даними свого формату. В основному використання інформації, що зберігається в бібліотеці даних, носить довідковий характер.

Бібліотеки узагальнених операцій і бібліотеки даних до підключення їх до системи фізично не пов'язані між собою. Бібліотеки узагальнених операцій і даних формуються в єдину групу лише логічними зв'язками з урахуванням області їх застосування. Будучи підключеними до системи методи мають безпосередній вплив на бібліотеки даних. Внаслідок того, що команди узагальнених операцій є відкритими посиланнями, в процесі проектування проектувальник пов'язує їх з інформацією з бібліотек даних.

Команди узагальнених операцій мають формат подання, використовуваний керуючим процесором. Команди в керуючому процесорі записуються в змінну, що складається з n осередків. Перші два осередки є службовими, в них записуються:

1. ідентифікатор тематичного співпроцесора, якому призначена команд;
2. найменування команди виконуваної тематичним співпроцесором.

Решта $n-2$ осередки є вільними посиланнями, заповнюваними системою, або проектувальником в процесі роботи.

Безпосереднє використання самих файлів бібліотеки узагальнених операцій настає в той момент, коли проектувальник формує сценарій проектування. Проектувальник викликає бібліотеку. Система відкриває файл з узагальненими операціями і готує його до роботи. Далі у формований сценарій проектування переписується обрана узагальнена операція. Після формування сценарію проектування файл бібліотеки узагальнених операцій закривається і знаходиться в закритому стані до наступного виклику.

Файл бібліотеки узагальнених операцій взаємодіє з модулем ведення сценарію проектування керуючого процесора, який виробляє безпосередні маніпуляції з файлом бібліотеки узагальнених операцій. Також файли бібліотек узагальнених операцій взаємодіють з ТП. У процесі зміни складу бібліотек менеджер бібліотек дає команди ТП для обробки інформації в файлах.

Безпосереднє управління бібліотекою узагальнених операцій здійснює модуль ведення сценарію проектування керуючого процесора. Він виконує команди проектувальника, що виробляють дії з файлами бібліотеки. У момент редагування складу бібліотек узагальнених операцій управління передається менеджеру бібліотек, який за допомогою ТП виробляє редагування інформації у файлі бібліотеки.

Бібліотеки узагальнених операцій можуть бути відредаговані. У випадках, коли проектувальнику видається недостатнім використання стандартних бібліотек узагальнених операцій, надається можливість створювати свої власні бібліотеки узагальнених операцій. Середовище дозволяє створювати абсолютно нові бібліотеки узагальнених операцій і доповнювати існуючі. Деякі бібліотеки,

що представляють певну цінність для роботи середовища, можуть бути позначені що не редаговані. Внаслідок чого узагальнені операції цих бібліотек захищені від необережного зміни.

Бібліотека узагальнених операцій складається з реєстраційного файлу бібліотеки узагальнених операцій і декількох файлів узагальнених операцій (рис. 3.15). Реєстраційний файл бібліотеки узагальнених операцій призначений для зберігання інформації про файли узагальнених операцій. У нього заноситься ім'я файлу узагальненої операції, включеного до бібліотеки, і шлях доступу до нього.



Рис. 3.15. Структура бібліотек узагальнених операцій

Файл узагальнених операцій має розвинену структуру (рис. 3.16). До його складу входять реєстраційний файл узагальнених операцій і файл набору узагальнених операцій. У реєстраційному файлі узагальнених операцій зберігається ім'я файлу набору узагальнених операцій, список найменування узагальнених операцій та адресу його розташування у файлі набору узагальнених операцій. Реєстраційний файл узагальнених операцій призначений для впорядкування інформації про наявні узагальнені операції і прискорення доступу до самим узагальненим операціями.

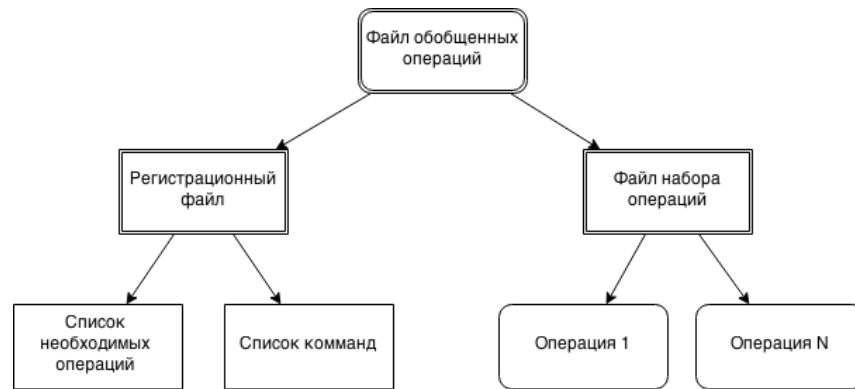


Рис. 3.16. Структура файла узагальнених операцій

Файл набору узагальнених операцій містить у собі операції, які представлені у вигляді записаних послідовно один за одним команд у форматі керуючого процесора. У файлі набору узагальнених операцій немає певних меж окремих операцій. Для зручності пошуку і прискорення обробки координати початку і кінця узагальненої операції записуються в реєстраційний файл узагальнених операцій. Команди, що входять до складу узагальненої операції, зберігаються у файлі послідовно один за одним.

Бібліотека даних складається з реєстраційного файлу бібліотеки даних і набору файлів даних (рис. 3.17). Реєстраційний файл бібліотеки даних призначений для зберігання інформації про файли даних. У нього заноситься ім'я файлу даних включеного до бібліотеки, шлях доступу до нього і приналежність файлу до того чи іншого тематичним співпроцесору.

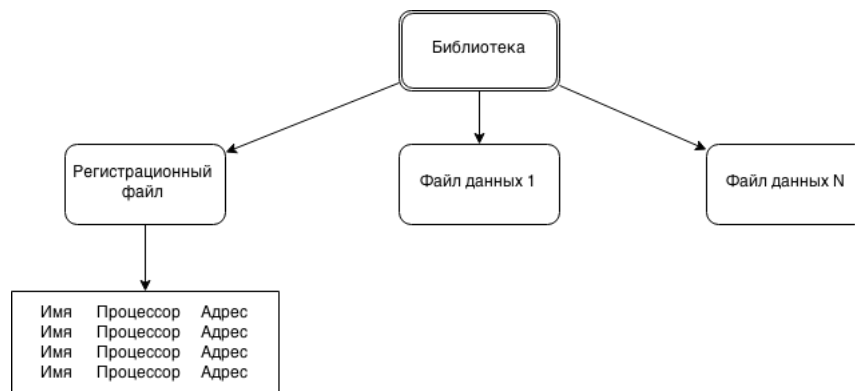


Рис. 3.17. Структура бібліотек даних

Файл даних має розвинену структуру. До його складу входять реєстраційний файл даних і файл набору даних. У реєстраційному файлі даних зберігається ім'я файлу набору даних, список найменування груп даних та адресу їх розташування у файлі набору даних. Реєстраційний файл даних призначений

для впорядкування інформації про наявні дані і прискорення доступу до самих даних.

Менеджер бібліотек

Використання в середовищі автоматизованого проектування бібліотек узагальнених операцій і даних викликало необхідність створення Менеджера бібліотек. На Менеджер бібліотек покладаються функції створення, відстеження та перекомпонування бібліотек. Менеджер бібліотек являє собою окремий програмний компонент середовища автоматизованого проектування.

Для управління діями процесорів Менеджер бібліотек має кілька функцій:

- створення бібліотек;
- перекомпонування бібліотек;
- відстеження стану бібліотек

Бібліотеки даних і узагальнених операцій формуються шляхом зв'язування файлів даних і файлів узагальнених операцій в єдиний комплекс. Проводиться заповнення всіх реєстраційних файлів, присутніх в бібліотеці. Для поліпшення якості бібліотек найбільш доцільним представляється використовувати існуючі електронні довідники. Процес створення бібліотек зводиться до виконання ряду дій:

- створення та заповнення реєстраційних файлів;
- при необхідності, приведення файлів до певного формату;
- компоновка бібліотеки.

Менеджер бібліотек є невід'ємною частиною середовища автоматизованого проектування. У процесі проектування часто виникає необхідність зробити певні дії з використовуваними бібліотеками. Використовувати менеджер бібліотек одночасно з процесом проектування можна. Дозволяється запускати Менеджер бібліотек тільки при зупиненому процесі проектування, тобто тільки при формальних ознаках: Файл сценарію проектування вивантажений і файли бібліотек не зайнята транзакціями.

Драйвери

Драйвери являють собою сполучну ланку між тематичним співпроцесором і виконуючим процесором. Драйвери вирішують завдання транслювання команд тематичних співпроцесорів в команди виконуючого процесора і дозволяють вибрати найбільш ефективний спосіб зв'язку з виконуючим процесором. Кожен виконуючий процесор оперує командами власного формату. Формат команд виконуючого процесора залежить від способу підключення виконуючого процесора до тематичного співпроцесору.

Більшість виконуючих процесорів має розвинену структуру вбудованої мови. Найбільш поширені програми мають два типи уявлення вбудованої мови:

- командна мова - мова, що має строкове представлення операторів і змінних;
- об'єктна мова - мова, що представляє собою набір об'єктів (класів) включають в себе методи і функції.

Є можливість використовувати кілька стандартних типів зв'язку додатків надається системою Windows:

- DDE - динамічний обмін даними, що дозволяє зраджувати і приймати дані;
- OLE - впровадження та зв'язування об'єктів - тип зв'язку, що дозволяє прив'язувати дані однієї програми до об'єкта в іншому додатку;
- Технологія COM і ActiveX - дозволяє отримати повне управління як окремим об'єктом додатку, так і самим додатком в цілому і проводити зміни у зовнішньому додатку;
- ODBC, DBE і SQL - засоби доступу до баз даних.

У кожному окремо взятому виконуючому процесорі структури командних та об'єктних мов різні. У більшості виконуючих процесорів найбільш розвиненим є об'єктний мову. Об'єктний мову видається більш зручним для його автоматизованого використання. Внаслідок чого найбільш доцільно використовувати об'єктна мова для зв'язку з тематичним співпроцесором другого рівня. Але в деяких виконуючих процесорах більш розвиненим представляється командна мова, і використання його є більш ефективним.

У середовищі автоматизованого проектування застосовується вбудований процесорний мову об'єктного типу. Використання драйверів тільки для мов

об'єктного типу, виходячи з вище сказаного, є неефективним. Тому потрібно ввести додатковий тип драйверів транслюють команди об'єктного процесорного мови в командний мову виконуючого процесора. Таким чином, застосування драйверів двох типів підвищує гнучкість і швидкодію системи.

Наступне завдання, яке вирішується драйвером, являє собою вибір найбільш ефективного способу зв'язку тематичного співпроцесора з виконуючими процесорами. Найзручнішим і ефективним способом зв'язку є використання технології COM і ActiveX. Але істотним недоліком цієї технології є задіяння великої кількості ресурсів комп'ютера для підтримки зв'язку. Оскільки в системі використовуються драйвери командного і об'єктного мов, то використовувати для них вимагається відповідні засоби зв'язку.

Для об'єктного мови оперує об'єктами доцільно використовувати технології COM і ActiveX як засіб зв'язку об'єктного рівня. Для командного мови, що оперує в основному дискретними даними, найбільш ефективним за витратами ресурсів є застосування технології динамічного обміну даними DDE.

Таким чином структура драйвера являє собою вбудований об'єктно орієнтований модуль. У драйвері для кожної команди тематичного співпроцесора є об'єкт-аналог для виконуючого процесора. При передачі команди у виконуючий процесор модуль обробки вихідних команд вибирає в драйвері об'єкт-аналог і запускає його виконання. Таким чином, необхідна команда виконується виконуючим процесором.

Для драйверів командного мови застосовується структура подібна драйверам об'єктного мови. Відмінністю є спосіб виконання об'єкта-аналога. У об'єкта-аналога поміщається код, що генерує текстовий рядок. При надходженні команди модуль обробки вихідних команд зраджує дані в драйвер. Драйвер, згенерував текстовий рядок, передає її в модуль зв'язку третього рівня. Модуль зв'язку третього рівня через підтримувану DDE-зв'язок передає отриману текстовий рядок на виконання виконуючому процесору.

Таким чином, використовуючи різний зміст об'єкта-аналога в драйвері, досягається сумісність об'єктного мови тематичних співпроцесорів з різними типами мов у виконуючому процесорі.

Файли драйверів являють собою динамічні бібліотеки DLL підключаються до тематичних співпроцесорами. Такий спосіб реалізації драйверів представляється найбільш доцільним при використанні об'єктів-аналогів. Інші способи реалізації драйверів показали гірші результати за різними параметрами. Наприклад, спосіб використання одного універсального об'єкта-аналога і файлів-наборів параметрів викликає збільшення кількості файлів що завантажуються, внаслідок чого підвищується відсоток помилок транслявання команд. Також знижується ефективність внесення змін в нових версіях драйверів, оскільки більшу кількість файлів знижує швидкість створення драйвера.

При дослідженні роботи механізму драйвера були показані добрі результати, достатні для ефективного функціонування системи.

У цьому розділі продемонстрована можлива реалізація програмних засобів, що забезпечують динамічну інтеграцію даних по запропонованому реляційному методу в середовищі автоматизованого проектування. Показана можливість створення серверного вузла середовища автоматизованого проектування, що виконує функції керуючого процесора. Продemonстровано можлива реалізація засобів динамічної інтеграції, що виконують функції тематичних співпроцесорів. Розроблено допоміжні програмні засоби, що полегшують роботу з середовищем автоматизованого проектування. Представлена можлива програмна реалізація середовища автоматизованого проектування демонструє реалізованість запропонованого реляційного методу динамічної інтеграції даних.

Висновки до розділу

Ми продемонстрували можливість розробки програмних засобів, що забезпечують динамічну інтеграцію даних по запропонованому реляційному методу в середовищі автоматизованого проектування. У представлений реалізації виконується повне відпрацювання всіх узагальнених операцій, наявних в сценарії проектування. Розроблено структурну схему Керуючого процесора. Реалізовано механізм передачі даних між процесорами.

Розроблено структурну схему тематичних співпроцесорів і можливі команди і повідомлення, якими обмінюються процесори. Розроблено драйвери, призначені для трансляції команд тематичних співпроцесорів в команди Виконуючих процесорів. Розроблено структуру Виконуючих процесорів, які є програмними засобами, що обробляють дані відповідних аспектів представлення. Розроблено надбудова для Виконуючих процесорів, що дозволяє забезпечувати зв'язок з тематичними співпроцесорами.

Розроблено структуру прикладних бібліотек, що включає в себе бібліотеки узагальнених операцій і бібліотеки методів. Прикладні бібліотеки являють собою пов'язані набори структурованих файлів, в яких зберігаються узагальнені операції і посилання файли даних різних аспектів уявлення. Розроблено Менеджер бібліотек, який забезпечує створення, редагування, і відстеження стану бібліотек.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

У цьому розділі розглядаються умови працівника під час роботи за електронно-обчислювальної машини при використанні програмного забезпечення, яке проектувалось під час написання дипломної роботи.

Передбачається, що працівник буде взаємодіяти безпосередньо з персональним комп'ютером, на якому буде встановлене необхідне програмне забезпечення в офісному приміщенні.

При роботі з персональним комп'ютером користувач відчуває значні навантаження на зір, м'язові навантаження, піддається електромагнітному випромінюванню, в зв'язку з переробкою великого обсягу інформації у користувача настає стомлюваність, що призводить до зниження працездатності і збільшення кількості помилок у виконуваний роботі. Так само негативно впливають: неправильне освітлення, неправильне планування робочого місця, погана вентиляція і обігрів приміщення.

Таким чином, при організації робочого місця важливо приділити увагу питанням охорони праці. Від умов роботи працівників залежить швидкість та якість роботи.

4.1 Характеристика виробничого приміщення

Досліджуваним робочим приміщенням є кімната (рис.4.1), що містить вісім робочих місць програмістів.

Загальна площа приміщення складає 51 квадратний метр, а загальний об'єм – 163,2 метрів кубічних. Висота стелі – 3,2 метри. Площа на одне робоче місце становить 6,375 м², а об'єм – 20,4 м³.

Робочі столи виготовлені з ДСП та мають два метри завдовжки та один метр завширшки. Сидіння робочих місць відповідають ергономічним вимогам для найбільш зручного положення тіла при роботі з ВДТ. Відстань між користувачем та ВДТ складає приблизно 700 мм. Висота робочої поверхні столу становить 750 мм.

Відеодисплейний термінал на кожному з робочих місць розташований на відстані 600 мм від краю поверхні стола. Окрім терміналу, кожне робоче місце містить власне комп'ютер, клавіатуру та мишу.

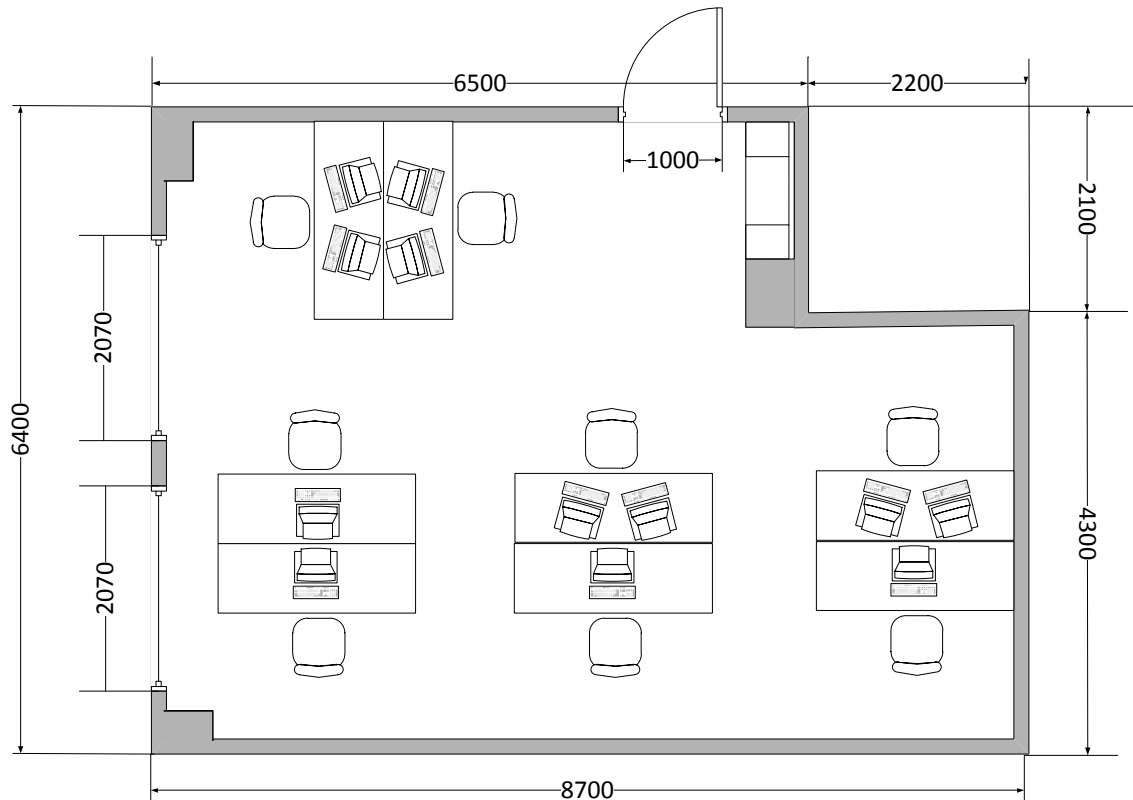


Рис.4.1. План приміщення

4.2 Ергономічна безпека

Правильна організація робочих місць запобігає передчасній втомлюваності користувача і сприяє збереженню здоров'я. Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні. Робочі місця у приміщенні групуються по 2 стола та знаходяться на відстані 1 м одне від одного. Площа на одне робоче місце становить не менше ніж $6,375 \text{ м}^2$, а об'єм – не менше ніж $20,4 \text{ м}^3$. Загальна площа приміщення складає 51 квадратний метр, а загальний об'єм – 163,2 метрів кубічних. Висота стелі – 3,2 метри.
- вибір ергономічно обґрунтованого робочого положення, виробничих меблів з урахуванням характеристик людини. Робочі столи виготовлені з ДСП та мають два метри завдовжки та один метр завширшки. Сидіння робочих місць відповідають ергономічним

вимогам для найбільш зручного положення тіла при роботі з ВДТ: тулуб розташований вертикально, кут, що утворюється між тулубом та стегновими кусками, складає 90 градусів, кут між стегном та гомілкою складає 90-110 градусів, кут між тулубом та передпліччям складає приблизно 0 градусів. Руки лежать на столі, відстань між користувачем та ВДТ складає приблизно 700 мм. Висота робочої поверхні столу повинна регулюватися у межах 680—800 мм; у середньому вона становить 725 мм.

- раціональне компонування обладнання на робочих місцях. Відеодисплейний термінал на кожному з робочих місць розташований на відстані 600 мм від краю поверхні стола. Окрім терміналу, кожне робоче місце містить власне комп'ютер, клавіатуру та мишу.

- урахування характеру й особливостей трудової діяльності. Організація робочого місця користувача ВДТ має забезпечуватися відповідно до НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин».

Що стосується умов праці користувачів ВДТ, пов'язаних з гігієною площі і виробничих приміщень, то вона також має відповідати НПАОП 0.00-1.28-10, згідно з якими площа, де має розташовуватися одне робоче місце з ВДТ, має становити не менш ніж 6,0 м², а об'єм приміщення бути не менше 20,0 м³.

4.3 Мікроклімат виробничого приміщення

Згідно ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень», нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця операторів ПК, встановлені оптимальні параметри мікроклімату, а при неможливості їх дотримання використовують допустимі параметри. Робота за персональним комп'ютером по енерговитратами відноситься до категорії легких Іб. В таблиці 6.1. наведені оптимальні

параметри мікроклімату в приміщеннях, де виконуються роботи операторського типу.

Таблиця 4.1. Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Величина
Холодний	Температура повітря в приміщенні Відносна вологість Швидкість руху повітря	21...23°C 40... 60% до 0,1 м/с
Теплий	Температура повітря в приміщенні Відносна вологість Швидкість руху повітря	22...24°C 40...60% 0,1...0,2 м/с

Виміряні за допомогою приладів температура та вологість у приміщенні відповідають вказаним у таблиці для теплого періоду року.

Слід зазначити, що для нормалізації параметрів мікроклімату слід використовувати у приміщеннях кондиціювання повітря, або забезпечити подачу свіжого повітря системами вентиляції. Норми подачі свіжого повітря наведені у табл. 4.2.

Таблиця 4.2 Норми подачі свіжого повітря в приміщення з ПК

Характеристика приміщення	Об'ємна витрата свіжого повітря, що подається в приміщення, м ³ на одну людину в годину
Об'єм до 20м ³ на людину 20... 40 м ³ на людину Більше 40 м ³ на людину	Не менше 30 Не менше 20 Може біти використана природна вентиляція

4.4 Освітлення

Наведемо для даного робочого місця характеристики зорових робіт :

- мінімальний розмір об'єкта розпізнавання – більше 5 мм; (мінімальний об'єкт розпізнавання в даній кімнаті є комп'ютер, текст на моніторах);
- контраст між об'єктом розпізнавання і фоном – є досить різний, оскільки він залежить від кольорів на лінії зору монітора комп'ютера;
- характеристику фону – також є різний;
- відносна тривалість зорової роботи в напрямку зору на робочу поверхню – менше 70%.

Визначення розряду та підрозряду робіт буде згідно таблиці 4.2 :

1. Розряд зорової роботи – середньої точності (оскільки об'єкт розпізнавання більше 0.5 мм);
2. Підрозряд – 2 (оскільки відносна тривалість зорової роботи в напрямку зору на робочу поверхню складає менше 70%);
3. Система природного освітлення – бокова;
4. Система штучного освітлення – загальна;
5. КПО для природного освітлення $e_n = 2\%$;
6. Нормоване значення освітленості для загальної системи штучного освітлення $E_n = 150$ лк.

Роботи в даному приміщенні ведуться у світлий час доби при природному освітленні, у темне – при штучному.

Природне освітлення здійснюється через два віконних прорізи розмірами 1,5х1,6 м та 1,4х1,6 м.

Одне вікно робочого приміщення виходить на південь, інше на північний схід.

Тип природного освітлення – бічне, одnobічне.

Розрахуємо нормативний коефіцієнт природної освітленості для робочого приміщення.

$$e_n = e_n \cdot m_N, \quad (4.1)$$

де $e_n = 0.5$ – значення КПО за таблицями 1 і 2 державних будівельних норм[18],

$m_N = 0.85$ – коефіцієнт світлового клімату за таблицею 4 державних будівельних норм [2];

N – номер групи забезпеченості природним світлом за таблицею 4 державних будівельних норм [2].

$$e_n = 0.5 * 0.85 = 0.425\%$$

Перевіримо відповідність розрахованого значення e_n дійсному $e_{\text{дійсн}}$ за допомогою приблизного проектного розрахунку.

Знайдемо світловий коефіцієнт природного освітлення за формулою:

$$e_{\text{дійсн}} = \frac{S_{\text{ВІК}}}{S_{\text{П}}} * 100, \quad (4.2)$$

де $S_{\text{ВІК}}$ – площа вікон, м^2 ;

$S_{\text{П}}$ – площа підлоги, м^2 .

$$e_{\text{дійсн}} = \frac{S_{\text{ВІК}}}{S_{\text{П}}} = \frac{1,5*1,6+1,4*1,6}{18} * 100 = \frac{4,64}{18} * 100 = 25,7\%$$

Оскільки 25,7% більше за 0,425 %, то природного світла достатньо для роботи в розглянутому приміщенні у світлий час доби.

В відповідності з [2], якщо в приміщенні не вистачає до норми природного освітлення та для освітлення в темний період доби, то воно повинне бути доповнене штучним.

Аналіз штучного освітлення

Для штучного освітлення використовуються люмінесцентні лампи – типу ЛБ 20, 8 штук. По чотири лампи в одному світильнику, всього два світильники.

Виконаємо аналіз штучного освітлення, який полягає у визначенні фактичної освітленості ($E_{\text{ф}}$), що створює система загального штучного освітлення, у порівнянні її з нормованим значенням E_n .

Для визначення фактичної освітленості скористаємося методом коефіцієнта світлового потоку, основне розрахункове рівняння якого має такий вигляд :

$$E_{\phi} = \frac{F_{\text{л}} N n \eta}{S k_3 z}, \quad (4.3)$$

де E_{ϕ} – нормована освітленість, лк;

$N = 2$ – кількість світильників;

$n = 4$ – кількість ламп у світильнику;

$\eta = 0.53$ – коефіцієнт використання світлового потоку;

$S = 51 \text{ м}^2$ – площа приміщення;

$k_3 = 1.3$ – коефіцієнт запасу;

$z = 1.15$ коефіцієнт нерівномірності освітлення;

$F_{\text{л}} = 1020 \text{ лк.}$ – світловий потік лампи.

$$E_{\phi} = \frac{1020 \cdot 2 \cdot 4 \cdot 0.53}{18 \cdot 1.3 \cdot 1.15} = 160 \text{ лк} > E_{\text{н}} = 150 \text{ лк.}$$

Отже, визначивши і порівнявши фактичну освітленість (E_{ϕ}) з нормованим значенням $E_{\text{н}}$, яке визначається згідно державним будівельним норми України[2], було визначено що штучного освітлення достатньо.

4.5 Характеристика випромінювання

Усе обладнання, що використовується в даному приміщенні, відповідає нормам, щодо електромагнітного випромінювання.

4.6 Аналіз рівня шуму

Можливими джерелами шуму в даному приміщенні є кулери процесорів, клавіатур. Сумарний рівень шуму в приміщенні складає близько 48дБА(з них 45.9дБА – шум від кулерів процесорних блоків, 2.1дБА – шум від клавіатур). Згідно ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку», еквівалентний рівень шуму має становити менше ніж 50дБА, тобто рівень шуму в приміщенні нижче за допустиму норму.

4.7 Електробезпека

Приміщення відноситься до приміщень без підвищеної електробезпеки.

Приміщення живиться електричною енергією від централізованого трансформатора, який знаходиться на поверсі поза межами приміщення і понижує напругу до 220 В, змінним струмом частотою 50 Гц. Комп'ютери мають

власні блоки живлення, що перетворюють змінний струм мережі у постійний (5 В, 12 В). Потужність даних блоків живлення – 400 – 450 Вт. ПК у приміщенні відносяться до категорії I захисту від ураження електричним струмом, оскільки усі прилади при підключенні до електричної мережі яка заземляється, використовуючи систему TN. Тобто система, у якій живильні мережі мають глухе заземлення однієї точки струмопровідних частин джерела живлення, а електроприймачі і відкриті провідні частини електроустановки приєднуються до цієї точки за допомогою відповідно нейтрального і захисного провідників.

Корпуса всіх електричних пристроїв виготовляються з неструмопровідних матеріалів, а живлення здійснюється спеціальним кабелем, так щоб виключити ураження людини електричним струмом. Зазначені міри електробезпеки відповідають вимогам, пропонованим до побутових приладів відповідно до міжнародного стандарту CE.

У приміщенні електропроводка трифазна, чотирьохжильна, з подвійною ізоляцією. Всі кабелі сховані в коробах. Штепсельні розетки встановлені на висоті одного та 0,2 метри від підлоги. Вимикачі на стінах розташовані на висоті 1,2 метра від підлоги з боку ручки для відкривання дверей. Таким чином, вимоги НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин» виконано.

4.8 Пожежна безпека

В приміщенні, що аналізується, наявні важкогорючі (елементи ЕОМ та інші електричні пристрої, з'єднувальні кабелі) та горючі (дерев'яні столи, шафи, паркет з натурального дерева та паперова документація). Горючі гази та горючі пиловидні речовини в завислому стані, що складають значну вибухову та пожежну небезпеку, а також горючі рідини в даному приміщенні відсутні.

Приміщення, для якого проводиться аналіз стану охорони праці, згідно НАПБ Б.03.002-2007 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою» відноситься до категорії В, класу П-Па по пожежонебезпеці.

В якості заходів щодо забезпечення пожежної безпеки прийняті наступні:

- призначений відповідальний за пожежну безпеку приміщення;
- розроблено план евакуації людей;
- у приміщенні заборонено користуватися нагрівальними приладами, курити, користуватися відкритим вогнем;
- є первинні засоби пожежогасіння у приміщенні (3 вогнегасника ВВ-2);
- встановлена пожежна сигналізація.

Відповідно до НАПБ Б.03.001-2004 «Типові норми належності вогнегасників» у приміщенні встановлено 3 вогнегасники типу ВВ-2 для тушіння невеликих джерел займання і устаткування під напругою до 1000 В. Відстань між місцями розташування вогнегасників не повинна перевищувати 15 м.

Для вчасного попередження про пожежу у приміщенні встановлена порогова пожежна сигналізація з пасивними сенсорами диму. У разі спрацювання системи сигнал надходить до чергового у лікарні.

Можливим джерелом запалювання є короткі замикання, що виникають внаслідок неправильного монтажу або експлуатації ЕОМ, старіння чи пошкодження ізоляції.

Для запобігання пожежі проводять планову заміну пошкоджених чи занадто старих кабелів. Монтаж нових ЕОМ та їх оновлення і ремонт проводяться за встановленою документацією. Експлуатація дозволяється тільки після проходження працівниками інструктажу від спеціаліста.

Ширина дверей 1м, при нормі не менше 0.9м, висота проходу 2,2 м при нормі не менше 2м, ширина коридору 2м (для одностороннього розташування дверей ширина евакуаційного шляху до світла, рівна ширині коридору, яку зменшено на половину ширини дверного полотна, повинна бути не менше 1м), що задовольняє вимогам.

4.9 Охорона праці та безпеки в умовах надзвичайних ситуацій

На відстані 1100 м від будівлі де розміщене робоче місце оператора ЕОМ в наслідок аварії або необережного поводження, може статися вибух приблизно

400т пропану. Тому, необхідно розрахувати можливі наслідки та вжити необхідні заходи щоб мінімізувати їх.

Вихідні дані:

- 1) Відстань від офісу до місця аварії (вибуху) – 1100 м;
- 2) Маса пропану – 400 т;
- 3) Характеристики елементів будівлі:
 - будівля – зі збірного залізо бетону;
 - кабельні лінії – наземні;
 - контрольно-вимірювальна апаратура – наявна;
 - границі вогнетривкості несучих стін – 3 год;
 - границі вогнетривкості перегородок – 1 год.
- 4) Категорія виробництва з пожежної безпеки – В;
- 5) Щільність забудови об'єкту - 48%.

Розрахункова частина:

Зона I : $r_1 = 17,5 \cdot \sqrt[3]{400} = 128,94$ м $\Delta P_{\phi} = 1700$ кПа

Зона II: $r_2 = 1,7 \cdot r_1 = 1,7 \cdot 128,94 = 219,2$ м

$$\Delta P_{II} = 1300 \cdot (r_1 / R_0)^3 + 50 = 1300 \cdot (128,94 / 1100)^3 + 50 = 50,002$$

Висновок: об'єкт знаходиться за межами I та II зон, тобто у зоні повітряної ударної хвилі (зона III)

$$\Delta P_{\phi} = \frac{262}{\sqrt{1 + 7,66 \cdot 10^{-5} \cdot \frac{L^3}{Q}} - 1} = \frac{262}{\sqrt{1 + 7,66 \cdot 10^{-5} \cdot \frac{1100^3}{400}} - 1} = 17,471$$

$\Delta P_{\phi} = 17,471$ кПа

При цьому маємо, наступні показники:

- Ступінь руйнування будівлі – слабкі;
- Ступінь руйнування контрольно-вимірювальної апаратури – середні;
- Ступінь руйнування кабельних ліній – слабкі;
- Ступінь ураження людей від прямої дії УХ – відсутні;
- Ступінь вогнестійкості – I;
- Небезпечна кількість вибухової речовини для уникнення руйнувань будівлі - менше 150 т;
- Небезпечна кількість вибухової речовини для уникнення будь-яких руйнувань – менше 40 т;

- Характеристика руйнувань будівлі: спостерігаються руйнування заповнень дверних та віконних прорізів, зривання покрівлі даху.
- Характеристика ураження людей: можливе отримання легких травм внаслідок вибиття вікон та дверей при дії вибухової хвилі
- Очікувана пожежна обстановка: для виробництва категорії пожежної небезпеки В та ступеня вогнестійкості будівель – І, при надмірному тиску 17,471 кПа пожежі не передбачуються.

Висновки:

На відстані 1100 м від будівлі стався вибух пропану, що призвело до часткових пошкоджень будівлі, а саме вибиття вікон, дверей та зривання покрівлі даху, а також наявність потерпілих людей з легкими травмами. В першу чергу необхідно надати першу медичну допомогу постраждалим та викликати швидку медичну допомогу.

Рекомендації що до зменшення заповіданої шкоди та уражень людей

- установити на вікнах захисні металеві сітки, щоб розбите скло не потрапляло в приміщення;
- порушити питання перед відповідними органами про зменшення запасу вибухонебезпечної речовини до безпечної кількості.

Висновок до розділу

Виходячи із даного розділу, можна відзначити, що приміщення, яке використовується програмісти для створення програмного забезпечення, має достатню ізоляцію, хороше освітлення. Для забезпечення пожежної безпеки виконується всі необхідні дії, по її запобігання, а також має достатню кількість вогнегасників у випадку непередбачених ситуацій. Приміщення відповідає всім вимогам, які висуваються стандартами.

ЗАГАЛЬНІ ВИСНОВКИ

У висновку представленої магістерської дисертації можна відзначити, що поставлене завдання досліджень, що припускає розробку інформаційної моделі середовища автоматизованого проектування і розробку реляційного методу динамічної інтеграції даних та засобів її реалізації, вирішена.

У процесі роботи розглянуто використання даних різних аспектів представлення із застосуванням засобів автоматизації проектування на основних етапах розробки та експлуатації об'єктів. При використанні засобів автоматизації проектування проектна документація інтерпретується у вигляді чотирьох аспектів представлення даних: графічний, табличний, текстовий, математичний. Розглянуто різні програмні засоби, що обробляють дані різних аспектів представлення, які використовуються для автоматизації проектування. Встановлено, що у всіх розглянутих програмних засобах реалізовані можливості для виконання інтеграції даних. Розглянуто різні методи і засоби інтеграції даних в САПР. Встановлено, що всі використовувані САПР мають можливість реалізації автоматизації проектування на рівні операцій з декларативними даними. Є деякі можливості для реалізації динамічної інтеграції даних графічного аспекту представлення за допомогою механізмів параметризації об'єктів або інтегрованого середовища САПР. Дані механізми не виходять за рамки графічного аспекту представлення даних. В рамках графічного аспекту реалізуються найзагальніші способи інтеграції даних. У розглянутих САПР динамічна інтеграція даних різних аспектів представлення не реалізована.

Для вирішення поставлених завдань запропоновано інформаційну модель середовища автоматизованого проектування, що включає в себе Керуючий процесор і чотири тематичні співпроцесори: Графічний співпроцесор, Табличний співпроцесор, Текстовий співпроцесор і Математичний співпроцесор. Введено поняття проекту, узагальненої операції, сценарію проектування і монітора, які використовуються як засоби формування інформаційної моделі середовища автоматизованого проектування. Розроблено схему та порядок взаємодії тематичних співпроцесорів і Керуючого процесора. Розроблено способи подання та взаємодії даних в інформаційній моделі

середовища автоматизованого проектування. Запропоновано розділяти відносини між даними на такі види: транзитивні, не транзитивні, виняткові. Розглянуто можливі види відносин між даними як одного, так і різних аспектів представлення. Встановлено, що не всі програмні засоби підтримують дані види відношень повністю. Запропоновано використовувати отриманий набір можливих відношень як формальне обмеження для середовища автоматизованого проектування. Запропоновано реляційний метод динамічної інтеграції даних в середовищі автоматизованого проектування, що полягає у використанні в процесі проектування узагальнених операцій, що складаються з команд різних тематичних співпроцесорів і інтегрованих даних як одного, так і різних аспектів представлення.

На основі запропонованої інформаційної моделі середовища автоматизованого проектування та реляційного методу динамічної інтеграції даних продемонстрована можливість розробки програмних засобів реалізації методу динамічної інтеграції даних в середовищі автоматизованого проектування. Розроблено структурну схему Керуючого процесора. Реалізовано механізм передачі даних між процесорами. Розроблено структурну схему тематичних співпроцесорів і можливі команди та повідомлення, якими обмінюються процесори. Розроблено драйвери, призначені для трансляції команд тематичних співпроцесорів в команди Виконуючих процесорів. Розроблено структуру Виконуючих процесорів, які є програмними засобами, які обробляють дані відповідних аспектів представлення. Розроблена надбудова для Виконуючих процесорів, що дозволяє забезпечувати зв'язок з тематичними співпроцесорами. Розроблено структуру прикладних бібліотек, що включає в себе бібліотеки узагальнених операцій і бібліотеки методів. Розроблено Менеджер бібліотек, який забезпечує створення, редагування, і відстеження стану бібліотек.

Для підтвердження перспективи застосування реляційного методу і розроблених засобів динамічної інтеграції даних проведена апробація стосовно до умов реального проектування. Підтверджено придатність методу для вирішення завдань проектування систем моніторингу розосереджених

програмних засобів. Проведена апробація методу динамічної інтеграції даних з використанням розробленого програмного комплексу для проектної організації, що займається проектуванням систем освітлення. Підтверджено придатність методу та засобів її реалізації стосовно до умов реального проектування. Зроблена оцінка прикладних бібліотек для використання в середовищі автоматизованого проектування для різних умов проектування, як у навчальному процесі, так і в умовах реальної проектної організації. Наведено основні критерії формування прикладних бібліотек. Підтверджено придатність прикладних бібліотек для застосування, як у навчальному процесі, так і в умовах реального проектування.

В результаті проведених досліджень отримані наступні основні результати:

1. Проведено аналіз інтеграції даних і програмних засобів автоматизації на основних етапах процесу проектування. Показано, що автоматизація проектування являє собою набір пов'язаних проектних даних, виражених різними аспектами представлення, встановлено:

- в процесі проектування використовуються дані графічного, табличного, текстового і математичного аспектів представлення;
- проектні дані поділяються на декларативну і процедурну складові;
- більшість програмних засобів, що входять до складу середовища автоматизованого проектування, мають можливість обробляти як декларативні, так і процедурні складові проектних даних.

2. Запропоновано нову інформаційну модель середовища автоматизованого проектування, що забезпечує динамічну інтеграцію даних різних аспектів представлення.

3. Запропоновано реляційний метод динамічної інтеграції, що представляє собою взаємодії засобів автоматизації проектування, що дозволяє реалізувати динамічний зв'язок проектних даних і відображення динамічних процесів модифікації методів і умов проектування, який заснований на побудові зв'язків між процедурними складовими проектних даних.

4. Розроблена вдосконалена модель узагальненої операції як складової сценарію проектування, що реалізує динаміку інтеграції даних різних аспектів подання в єдиному інформаційному процесі.

5. На основі розробленого комплексу програмних засобів перевірена на практиці реалізація запропонованого реляційного методу динамічної інтеграції даних в середовищі автоматизованого проектування. При апробації встановлено, що на основних етапах реального процесу проектування промислових об'єктів:

- використання узагальнених операцій дозволяє підвищити ефективність виконання однотипних операцій;
- використання запропонованого реляційного методу динамічної інтеграції даних при виконанні проекту дозволяє спростити процес коригування та модифікації проекту;
- використання існуючих прикладних бібліотек дозволяє підвищити ефективність застосування реляційного методу динамічної інтеграції даних в середовищі автоматизованого проектування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. СНиП 2.04.05-92. Отопление, вентиляция и кондиционирование воздуха.
2. ДБН В.2.5.-28-2006. Природне і штучне освітлення. Мінбуд України, 2006 - 76 с.
3. СН 181-70 Указания по проектированию цветовой отделки интерьеров производственных зданий промышленных предприятий.
4. ДСанПіН 3.3.2.007-98 Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
5. ТСО 5.1 Стандарты эргономичности и безопасности электронного оборудования.
6. ДСН 3.3.6.042-99 Державні санітарні норми мікроклімату виробничих приміщень.
7. ДСН 3.3.6.037-99. Санітарні норми виробничого шуму, ультразвуку та інфразвуку.
8. ДСН 3.3.6.039-99. Санітарні норми виробничої загальної та локальної вібрації.
9. НАПБ Б.03.002-2007. Нормы определения категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности.
10. СНиП 2.01.02–85. Строительные нормы и правила. Противопожарные нормы.
11. ГОСТ 12.1.044-89. ССБТ. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методика определения.
12. ДНАОП 0.00-1.31-99 «Правила охорони праці під час експлуатації електронно-обчислювальних машин».
13. НАПБ Б.03.002-2007. Нормы определения категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности.
14. СНиП 2.01.02–85. Строительные нормы и правила. Противопожарные нормы.
15. ГОСТ 12.1.044-89. ССБТ. Пожаровзрывоопасность веществ и материалов. Номенклатура показателей и методика определения.

- 16.ДНАОП 0.00-1.31-99 «Правила охорони праці під час експлуатації електронно-обчислювальних машин».
- 17.Норенков И.П. Принципы построения и структура САПР. — М.: «Высшая школа», 2001 г.
- 18.Клир Дж. Системология. Автоматизация решения системных задач. — М.: «Радио и связи». 1990 г.
- 19.Евгениев Г.Б., Основы инженерной системологии. МГТУ им. Н.Э. Баумана. Москва. 1998 г.
- 20.Правила технической эксплуатации электрических станций и сетей Российской Федерации. — 15-е изд., перераб. и доп. — СПб.: Деан, 2000. — 352с.
- 21.Нестеров Ю.Г., Папшев И.С., Выбор состава программно-технического комплекса САПР. — М.: «Высшая школа», 1990 г.
- 22.Федоров Б.С., Гуляев Н.Б., Проектирование программного обеспечения САПР. — М.: «Высшая школа», 1990 г.
- 23.Климов В.Е., Графические системы САПР. — М.: «Высшая школа», 1990г.
- 24.Вейнеров О.М., Самохвалов Э.Н., Проектирование баз данных САПР. — М.: «Высшая школа», 1990 г.
- 25.Штейнбрехер А., Чилингаров К. Использование существующих баз данных при внедрении автоматизированной системы подготовки производства. «САПР и Графика», № 4, 2001, 14-16 с.
- 26.Персон Р., Роуз К. Microsoft Word 97 в подлиннике / Пер. с англ. СПб.: ВНУ - Санкт - Петербург, 2001 - 120 с.
- 27.Дьяконов В. Mathcad 2000: Учеб. Курс. - СПб.: Питер, 2000. - 592 с. - ил.
- 28.Черненький В.М., Имитационное моделирование. — М.: «Высшая школа», 1990 г.
- 29.Куприянов В.В., Печенкин О.Ю., Суслов М.Л., Уколов И.С., САПР и системы искусственного интеллекта на базе ЭВМ. — М.: «Наука», 1991 г.
- 30.Антонов А., Мурованная Е. SolidWorks 2001 — верность традициям. «САПР и Графика», № 3,2001,29-35 с.

31. Антонов А., Мурованная Е. Разрешите представиться — SolidWorks 99. «САПР и Графика», № 1, 2000, 21-24 с.
32. Бикулов С, Козлов С, Баранов Д., Семин Д. Новые возможности T-FLEX CAD 7.1. «САПР и Графика», № 6, 2001, 58-60 с.
33. Дудка В., Морозов А. и др. Разведка боем: передовые технологии SolidWorks в ГУН «КБП» (г. Тула). «САПР и Графика», № 2, 2001, 82-88 с, «САПР и Графика», № 4, 2001, 30-33 с, № 5, 2001, 76-82 с.
34. Пирогов В., Пирогов И., Степанов А. Новые возможности T-FLEX CAD/CAM при обработке сложных деталей. «САПР и Графика», № 5, 2001, 73-75 с.
35. Беленький Ю., Власенко С, Microsoft® Word 2000. - СПб: BHV - Санкт - Петербург, 2000. - 967 с: ил.
36. Данчул А.Н., Полуян Л.Я., Системно-технические задачи создания САПР. — М.: «Высшая школа», 1990 г.
37. Карпова Т. Базы данных: Модели, разработка, реализация: Учеб. СПб.: Питер, 2001.-304 с: ил.
38. Конноли Т., Бегг К., Страчан А. Базы данных: Проектирование, реализация и сопровождение. Теория и практика. -2-е изд., испр., доп. / Пер. с англ. М.: Вильяме, 2000. -1120 с.
39. Старостина Л.А., Введение в AutoCAD. — М.: «Бутэк», 1991. —112с., ил.
40. Сигнор Роберт, Стегман Михаэль О., Использование ODBC для доступа к базам данных. Пер. с англ. — М: «Бином», 1995. — 384 с, ил.
41. Боуман Дж.С, Эмерсон СЛ., Дарновски М. Практическое руководство по SQL: использование языка структурированных запросов / Пер. с англ. -3-е издание. М.: Вильяме, 2001.-336 с.
42. Гультияев А. К. MATLAB 5.3: Имитационное моделирование в среде Windows. СПб.: КОРОНА принт, 2001.- 400 с.
43. Дьяконов В. Mathematica 4. - СПб.: Питер, 2001. - 656 с: ил.
44. Дьяконов В.П. Mathematica 4 с пакетами расширений. М.: Нолидж, 2000. - 608 с.

- 45.Аладьев В.З., Шишаков М. Л., Введение в среду МАТНЕМАТИСА 2.2 — М.: «Филинь», 1997. — 368 с.
- 46.Губич Л., Прохорова А. Подходы к автоматизации проектирования на базе средств параметризации САД/САМ-систем. «САПР и Графика», № 12, 2000, 60–64 с.
- 47.Ковалевский В. Системная интеграция с точки зрения СПРУТ-технологии. Информационная интеграция. «САПР и Графика», № 7,1998, 31-36 с.
- 48.Кузьмин Б., Лебедев С, Тагиев Д. Система проектирования технологических процессов компании «СПРУТ-технология» как основа интегрированной АСТПП. «САПР и Графика», № 10,1999, 50-55 с.
- 49.Ковалевский В., Савинов А. Методическая интеграция прикладных систем, разработанных с использованием интегрированной среды СПРУТ. «САПР и Графика», № 8,1998,28-34 с.
- 50.Джамп Д., AutoCAD. Программирование. Пер. с англ. — М.: «Радио и связь», 1992. —336 с, ил.
- 51.Роджерс Д., Адаме Дж. Математические основы машинной графики / Пер. со 2 - го англ. Изд. М.: Мир, 2001. - 604 с.
- 52.Локтев В. Система «КРЕДО» — управление технической документацией. «САПР и Графика», № 1,1999, 85-88 с.
- 53.Потапов А. WAVE — новая технология системы Unigraphics. «САПР и Графика», № 8, 1998, 63-65 с.
- 54.Интегрированный процесс проектирования. Работаем с программным обеспечением фирмы Немечек. «САПР и Графика», № 6, 2001, 95-98 с.
- 55.Хауз Р. Использование AutoCAD® 2000: Спец. изд. / Пер. с англ. - М.: Вильяме, 2000. - 832 с.
- 56.Бикулов С, Ксенофонтов Д. Параметризация в T-FLEX CAD. «САПР и Графика», № 4,1999,69-72 с.
- 57.Артемьев В.И., Строганов В.Ю., Организация диалога в САПР. — М.: «Высшая школа», 1990 г.

- 58.Новоселов В., Михайлов М., Холдин М. Интеграция с SolidWorks: программа для создания конструкторской спецификации. «САПР и Графика», № 2, 2000, 83-85 с.
- 59.Хараджиев А. Комплексная автоматизация подготовки производства на базе СПРУТ-технологии. «САПР и Графика», № 11, 2000, 41-44 с.
- 60.Сухов С.А. Инструментальная диалоговая Среда для создания прикладных (конечных) САПР. Диссертация, Ульяновск, УлГТУ, 1996 .
- 61.Семенов В.Г. Сетевые модели оперативного управления процессом принятия решений в САПР. Диссертация, Ульяновск, УлГТУ, 2001.
- 62.Kalabanovsky I.A., Pohilko A.F. Data presentation and criterion of investigation of CAD - systems for the class of problem with the structured presentation of data. // "Interactive systems: The problems of human - computer integration": 3-d International Conference - Ulyanovsk, 1999.
- 63.Pohilko A.F., Kalabanovsky LA. Dynamic Integration of Visual Data and Procedure Applications in the Systems of Computer-Aided Design. //Pattern Recognition and Image Analysis. Vol. 9,2. 1999.
- 64.Калабановский И.А. Представление данных в информационной модели динамической интеграции данных в системах автоматизированного проектирования. // статья, "Вестник УлГТУ", январь - март 1998 г.
- 65.Калабановский И.А. Создание приложений в интегрированной среде САПР. // тез. докл. XXXIII научно-технической конференции - Ульяновск, 1999г.
- 66.Калабановский И.А., Похилько А.Ф. Декомпозиция процесса проектирования систем электроснабжения по элементам интегрированной среды проектирования. // «Нейронные, реляторные и непрерывно-логические сети и модели» научно-техническая конференции тез. докл. Ульяновск, 1998 г.
- 67.Калабановский И.А., Похилько А.Ф. Динамическая интеграция данных в системах автоматизированного проектирования электросистем. // "Наука - производству: конверсия сегодня": тез. докл. научно - практической конференции - Ульяновск, 1997 г.

- 68.Калабановский И.А., Похилько А.Ф. Информационная модель динамического интегрирования данных в системах автоматизированного проектирования. // XXXII научно-техническая конференции тез. докл. Ульяновск, 1998 г.
- 69.Калабановский И.А., Похилько А.Ф. Модель динамической интеграции при создании процедурных приложений в САПР. // III Международная научно-техническая конференция «Новые информационные технологии и системы» тез. докл. Пенза, 1998 г.
- 70.Калабановский И.А., Похилько А.Ф. Структурированные приложения проектирования электросистем // XXXI Научно-техническая конференция, тез. докл. Ульяновск, 1997 г.
- 71.Калабановский И.А., Похилько А.Ф., Бондаренко Е.В. Метод динамической интеграции данных, применяемый в САПР СЭС сельских районов. // статья в журнале "Электрификация и механизация сельского хозяйства" - Москва, № 5, 1999 г.