

Національний університет "Львівська політехніка"

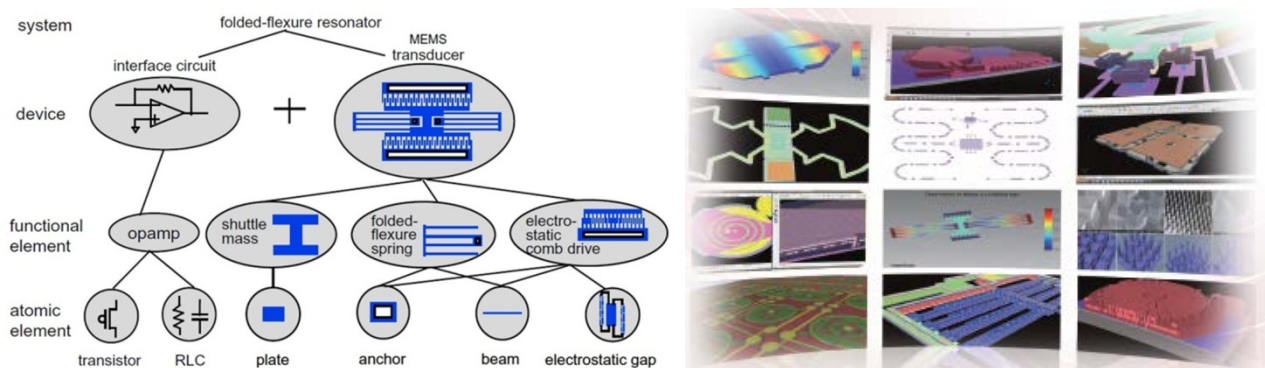
Харківський національний університет радіоелектроніки

Олег Матвійків, Сергій Ткаченко, Володимир Хаханов

Інженерне проектування складних об'єктів і систем Engineering Design of Complex Objects and Systems

Навчальний посібник

(draft version)



№530785-TEMPUS-1-2012-1-PL-TEMPUS-JPCR



Tempus

Навчальний посібник **"Інженерне проектування складних об'єктів і систем"** створено для допомоги вищим навчальним закладам України впровадити нову магістерську навчальну програму "Проектування мікросистем".

Посібник **"Інженерне проектування складних об'єктів і систем"** створено при підтримці Європейського Союзу за Спільним Європейським Проектом "Curricula Development for New Specialization: Master of Engineering in Microsystems Design" (MastMST), ідентифікаційний номер 530785-TEMPUS-1-2012-1-PL-TEMPUS-JPCR.

Координатор проекту проф. Збігнев Лісік, Технічний університет м.Лодзь, Польща.

Учасники проекту:

- Національний університет "Львівська політехніка", м. Львів, Україна, координатор проф. Михайло Лобур.
- Київський Національний університет ім. Тараса Шевченка, м. Київ, Україна, координатор проф. Валерій Скришевський.
- Харківський національний університет радіоелектроніки, м. Харків, Україна, координатор проф. Володимир Хаханов.
- Донецький національний технічний університет, м. Красноармійськ, Україна, координатор проф. Володимир Святний.
- Технічний університет м.Ільменау, Німеччина, координатор проф. Іво Рангелов.
- Ліонський Національний інститут прикладних наук, Франція, координатор проф. Александра Аполтолюк
- Університет Павії, м.Павії, Італія, координатор проф. Паоло Ді Барба

Посібник схвалено редакційним комітетом (проф. Паоло Ді Барба (Університет Павії) - співголова, проф. Александра Аполтолюк(Ліонський Національний інститут прикладних наук) – співголова, члени: проф. Збігнев Лісік(Технічний університет м.Лодзь), д-р Яцек Подгурські (Технічний університет м.Лодзь), Д-р Януш Возний (Технічний університет м.Лодзь), Д-р Валентин Іщук (Технічний університет м.Ільменау), Д-р Марія-Евеліна Могначі (Університет Павії), Д-р Роберто Галді (Університет Павії)) 6 травня 2016, м. Павія, Італія

Автори висловлюють глибоку вдячність керівництву вищеназваних університетів за всебічну підтримку Проекту.

Textbook "**Engineering Design of Complex Objects and Systems**" developed to help higher education institutions in Ukraine to introduce new master's educational program "Designing microsystems".

Textbook "**Engineering Design of Complex Objects and Systems**" was created with the support of the European Union within the Joint European Project "Curricula Development for New Specialization: Master of Engineering in Microsystems Design" (MastMST), identification number 530785-TEMPUS-1-2012-1-PL-TEMPUS-JPCR.

Project Coordinator prof. Zbigniew Lisik, Lodz University of Technology, Lodz, Poland.

Учасники проекту:

- Lviv Politechnical National University, Lviv, Ukraine ,
Coordinator prof. Mykhailo Lobur.
- Taras Shevchenko National University of Kyiv, Ukraine,
Coordinator prof. Valeriy Skryshevsky.
- Kharkiv National University of Radioelectronics, Ukraine ,
Coordinator prof. Vladimir Hahanov.
- Donetsk National Technical University, Krasnoarmiysk,
Coordinator prof. Volodymyr Sviatny.
- Ilmenau University of Technology, Germany,
Coordinator prof. Ivo Rangelow.
- Lyon Institute of Applied Sciences, France,
Coordinator prof. Alexandra Apostoluk.
- University of Pavia, Italy,
Coordinator prof. Paolo Di Barba.

The Handbook was approved by Editorial Committee (prof. Paolo Di Barba (University of Pavia) - Co-Chair, prof. Alexandra Apostoluk (Lyon Institute of Applied Sciences) – Co-Chair, members: prof. Zbigniew Lisik (Lodz University of Technology), Dr Jacek Podgorski (Lodz University of Technology), Dr Janusz Wozny (Lodz University of Technology), Dr Valentyn Ishchuk (Ilmenau University of Technology), Dr Maria Evelina Mognaschi (University of Pavia), Dr Roberto Galdi (University of Pavia) May 6, 2016, Pavia, Italy.

The authors express their deep gratitude to the aforementioned universities for full support of the project.

Анотація

Основною метою ТЕМПУС проекту було створення умов в українських технічних університетах для наскрізного 3-рівневого навчання спеціалістів в області автоматизованого проектування та інженерії мікросистем, відповідно до регіональних потреб ринку праці.

Цільова група: студенти, випускники, викладачі та адміністрація університетів, керівники промислових підприємств, міністерство освіти і науки України.

У навчальному посібнику розглядаються питання розробки і використання засобів автоматизації для проектування інженерних об'єктів і систем. Проведено аналіз існуючих процесів проектування елементів і технічних систем. Розглядаються різні підходи до проектування і даються типові проектні процедури.

Автоматизація побудови математичних моделей у САПР викладається з узагальнених позицій. Проводиться автоматизація аналізу і синтезу елементів технічних систем. Розглядаються різні методи синтезу й оптимізації. Викладаються аспекти конструкторського і технологічного проектування складних інженерних об'єктів засобами САПР.

Запропонований посібник буде корисним для студентів, аспірантів та науковців, які спеціалізуються в галузі автоматизованого проектування та математичного моделювання

Зміст

Зміст	5
Словник термінів і скорочень.....	7
Вступ.....	9
Характеристика навчального курсу	10
1. Загальні відомості про проектування.....	13
2. Поняття складного об'єкта чи системи.....	33
3. САПР в автоматизованому проектуванні СОС.....	45
4. Основні поняття та визначення автоматизованого проектування	69
5. Основні поняття та визначення в проектуванні.....	71
6. Маршрут автоматизованого проектування	83
7. Технологія паралельного проектування СОС	95
8. Системні середовища і програмно-методичні комплекси	103
9. Технічне забезпечення процесу проектування	123
10. Введення в пошукове проектування	136
11. Комп'ютерні методи пошукового конструювання	146
12. Евристичні методи пошукового конструювання нових технічних рішень.....	166
13. Роль комп'ютерного моделювання та прототипіювання.....	174
14. Компоненти математичного забезпечення (МЗ) АПСОС.....	180
15. ММ в процедурах аналізу на макрорівні	184
16. Математичне забезпечення аналізу на мікрорівні	191
17. ММ аналізу на функціонально-логічному рівні	195
18. Математичне забезпечення СОС на системному рівні	201
19. Математичне забезпечення підсистем машинної графіки і геометричного моделювання.....	211
20. Математичне забезпечення параметричного синтезу в АП СОС	217
21. Огляд методів оптимізації.....	220
22. Математичне забезпечення структурного синтезу в АПСОС	232
23. Методи структурного синтезу в САПР.....	237
24. Структура IntelliSuite	246
ЗАГАЛЬНА ЛІТЕРАТУРА	261

Словник термінів і скорочень

термін	англ
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAE	Computer Aided Engineering
PDM	Product Data Management

Department of Computer-Aided Design Systems at Lviv Polytechnic National University always invest all its efforts in first-rate facilities and equipment dedicated to teaching. TEMPUS project allowed to create two unique teaching laboratories - Microsystems Design Laboratory and Microsystems Testing laboratory, which support computer aided design (CAD) and computer aided engineering (CAE) by using industry-standard software packages and tools.

Actual course includes lectures, labs and individual 'hands-on' project, which includes investigations, experiments workshops and realisation of designs. These laboratory sessions and design project are the most enjoyable parts of teaching courses.

Abstract This course introduces students to fundamental topics in engineering design for research and practice covering the main methods, models, theory and methodology. The course will be taught using a number of case studies motivated by grand challenges in engineering design.

Participants will learn about the Computer-Aided Engineering fundamentals and methods that are necessary for successful design of modern technical products. The focus will be placed on the simulation-driven design in the context of product development process as well as on the fundamentals of the design optimization.

Objective The objectives of the course are to introduce students to the most important topics in design methods, models, theory and methodology that form the basis for engineering design practice and research. A further goal is to develop design reasoning and critical thinking skills.

Basic Computer-Aided Engineering (CAE) knowledge and skills will be acquired to enable students to recognize both the advantages and the limitations of current CAE tools. Examples of how to build feature-based and parametric models for simulation-driven design automation will be given along with common pitfalls. The CAE environment will be the Siemens NX 8.5 which couples the simulation modeling (e.g. structural, thermal, flow, motion, and multiphysics) with design optimization and Feature-Based Design (FBD). After taking the course students should be able to independently create effective feature-based and parametric models to suit the requirements of simulation-driven design.

Content The content of the course will be split into four units:

- 1) understanding designers,
- 2) design processes and practice and
- 3) products and designed artefacts.
- 4) Microsystems design with Intellisuite.

Within each unit key topics and methods will be covered including empirical design research, design science, creativity, processes for engineering design practice, user-centered design, re-design and reverse engineering, product models including functional modeling, product lifecycle and sustainability, design for manufacture including additive manufacturing, and integrated, networked products.

1. Computer-Aided Engineering (CAE) methods and tools in context of design process (2 afternoons): * CAE in the context of the design process; * Simulation-driven design; * Introduction to design optimization; * Features, parameterization and synchronous modeling technology; * Basic design optimization examples; * Introduction to Finite-Element Method (FEM) with basic examples.
2. Simulation-Driven Design with application to structural design (1 afternoon): * Coupling simulation with structural design optimization and feature based-design; * Simulation driven design examples (single parts and assemblies)

Lecture notes Handouts in the lecture, soon available in Moodle

Вступ

В процесі розвитку науки і техніки створювані людиною технічні системи, пристрої, структури стають все більш складними. Одночасно стають більш жорсткими вимоги до термінів проектування нових виробів. В цих умовах неавтоматизовані методи проектування виявляються неефективними. Створення і широке використання систем автоматизованого проектування (САПР) стає нагальною необхідністю.

Ефективність застосування засобів автоматизації багато в чому визначається своєчасною підготовкою кваліфікованих розробників і користувачів САПР. Користувач САПР обов'язково повинний не тільки володіти технологією проектування і мати навички роботи з готовими компонентами промислових САПР, але і вміти створювати прикладні програми для діючого програмного забезпечення САПР, що неможливо без врахування численних особливостей об'єкта проектування, та без знання сучасних технологій програмування.

Для рішення цієї важливої народногосподарської задачі необхідна відповідна підготовка спеціалістів. Випускникам вузів необхідно вміти працювати в якості кваліфікованих користувачів САПР. Крім користувачів необхідні також спеціалісти з проектування самих САПР. Сучасний фахівець повинний вміти використовувати ЕОМ на всіх етапах проектування: від постановки задачі на розробку складного об'єкта (або процесу) до їхнього приймання в експлуатацію. Тому в рамках різних інженерних спеціальностей передбачена підготовка інженерів-користувачів САПР що уміють вести автоматизоване проектування, конструювання і технологічну підготовку виробництва.

В наш час підготовка інженерів-проектувальників проводиться в рамках спеціальності «Інформаційні технології проектування» на бакалавраті «Комп'ютерні науки». Курс "Основи автоматизованого проектування складних об'єктів і систем" вивчається студентами-бакалаврами на останньому році навчання. Знайомство з цим курсом корисно не тільки тим фахівцям, що будуть працювати в проектних організаціях, але і усім фахівцям, що будуть зв'язані з промисловим виробництвом, а саме:

- керівників відділів та інших підрозділів САПР проектних інститутів, КБ,
- менеджерів проектів різних організацій,
- фахівців із програмного забезпечення,
- служб (відділів) САПР, АСТПП і інших.

Характеристика навчального курсу

Курс базується на дисциплінах:

Основи дискретної математики (теорія множин, теорія графів, алгебричні системи, теорія графіків, теорія автоматів, математична логіка).

Теорія ймовірностей (випадкові події, випадкові величини, граничні теореми, математична статистика).

Основи системного аналізу об'єктів та процесів комп'ютеризації (системний аналіз об'єктів і процесів, інформаційно-вартісний аналіз, цільовий аналіз, ситуаційний аналіз та аналіз поведінки об'єктів і процесів).

Організація баз даних і знань (методи організації БД, моделювання предметного середовища з різними моделями даних, організація реляційних БД, організація параметричних та розподілених БД, організація баз знань, алгоритмізація доступу та захист даних і знань).

Математичне моделювання в САПР

Прикладні проекти, розроблені студентами в рамках учбово-дослідницької роботи, надалі можна використовувати в ході дипломного проектування.

Мета і задачі дисципліни

Предметом вивчення дисципліни є методологія автоматизованого проектування; питання комплексування технічних засобів САПР; питання розробки і використання промислових систем автоматизованого проектування.

Цілі курсу:

- одержати представлення про цілях і способах автоматизації проектування,
- сформувані правильне розуміння ролі і місця засобів обчислювальної техніки в проектуванні і підготовці виробництва в сучасних умовах,
- виробити навички постановки задач на розробку або модернізацію елементів САПР.
- навчитися використовувати САПР при розробці нових пристроїв або їх модернізації і реконструкції, при рішенні оптимізаційних задач,

У результаті вивчення курсу студенти повинні **ЗНАТИ:**

- принципи системного підходу до проектування СОС;
- методологію автоматизованого проектування СОС (стратегію, технологію, етапи та маршрути проектування, типові проектні процедури);
- сценарії побудови дерева цілей, як метода цілеспрямованого проектування системи;
- склад та структуру промислових САПР, які застосовують для АП різноманітних СОС;
- принципи, методи та алгоритми проектування прикладних рутинних задач;
- постановку та методи рішення основних задач САПР – синтезу, аналізу, верифікації та оптимізації;
- методи побудови ММ ОП на мікро/мезо/макро та мета- рівнях;
- склад та структуру основних забезпечень САПР та методів їх використання для організації

10Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

У результаті вивчення курсу студенти повинні **ВМІТИ:**

- формувати завдання на проектування СОС в середовищі різних (гетерогенних) промислових САПР;
- розробляти ТЗ, техніко-економічне обґрунтування, технічний та робочий проекти в процесі проектування;
- організовувати процес АП СОС, формувати критерії ефективності автоматизації проектування;
- володіти методами побудови ММ об'єктів та систем, що проектуються; вміти об'єднувати та деталізувати їх;
- використовувати ММ при виконанні основних проектних процедур (аналізу, синтезу, верифікації, тощо);
- оцінювати ефективність застосування альтернативних елементів математичного забезпечення в САПР;
- збирати, зберігати, обробляти вихідні дані, виконувати багатоваріантні оптимізаційні розрахунки;
- вибирати раціональні варіанти та стійкі до варіації вихідних даних методи рішень задач при АП СОС;
- розробляти та комплексувати АРМ та інженерні станції;
- використовувати бази даних для зберігання, пошуку та аналізу проектної інформації;

Загальні методичні вказівки

Вивчення курсу "Основи САПР" здійснюється за допомогою наступних видів занять:

- лекції по основних темах курсу;
- лабораторні роботи (з використанням промислових САПР);
- самостійне вивчення студентами розділів курсу по літературі, що рекомендується;
- виконання семестрового завдання (графічно-розрахункова робота^{*}).

При самостійній роботі з літературою рекомендується вести короткий конспект досліджуваних питань. Контроль ступеня засвоєння матеріалу здійснюється за допомогою питань для самоперевірки, що охоплюють повний зміст методичних указівок по темах, і шляхом виконання домашнього завдання.

Семестрове завдання (ГРЗ) видається кожному студентові індивідуально. Робота являє собою розробку радіоелектронного пристрою. Можлива також розробка програмного забезпечення моделювання або розрахунку, зв'язаного з напрямком учбово-дослідницької роботи студента (за узгодженням з керівником).

Контрольна робота проводиться в кінці семестру і призначена оцінити отримані знання студентів перед заліком.

Частина I.

Методологія інженерного проектування

Техніка

Одна з найвідоміших у світі технічних енциклопедій, Mc Hill Encyclopedia of Science and Technology, визначає техніку як « мистецтво направляти і використовувати енергію природи на потреби та процвітання людства». Сучасна техніка охоплює людей, фінансові і матеріальні ресурси, машини та енергію. Головне, що відрізняє її від науки, полягає в тому, що першочерговим завданням техніки є пошук засобів корисного та економічно вигідного перетворення явищ природи, у той час як вчені відкривають ці явища і формулюють теорії для їх описання. Техніка завжди перебуває в пошуку нових дешевих і досконаліших засобів перетворення енергії і матеріалів, які націлені на підвищення рівня життя і полегшення праці людини. Людину, професійна діяльність якої пов'язана з технікою, зазвичай називають інженером, а саму професійну діяльність – інженерною діяльністю.

Слово «інженер» походить від латинського слова *ingeniosus* – обдарований, талановитий. Це визначення стосувалось людини, що постійно щось винаходила, винаходила. Так називали легендарного генія Леонардо да Вінчі, так називали нові, складні на перший погляд, дуже логічні і технологічно прості пристрої.

Коли Джеймс Уайт винайшов перший паровий двигун, його оцінили саме як такий хитромудрий винахідницький пристрій і назвали *ingenious*. Поступово, у міру поширення цих двигунів їх науку скоротили просто до «engine».

В одному з найновіших оксфордських словників зазначається, що «інженер – це той, хто проектує пристрої, механізми і споруди на користь суспільства». Як впливає з цього визначення, суттю інженерної діяльності, на відміну від діяльності наукової, є не накопичення знань про певні явища, не аналіз компонентів цих явищ, а саме проектування технічних систем в інтересах людини. Вчений вивчає те, що є; інженер створює те, чого не було. Тепер визначимось з тим, що ми будемо розуміти під технічними системами та їх проектуванням.

1. Загальні відомості про проектування

Поняття „інженерного проектування”

Проектування – це одна з найважливіших сфер інженерної діяльності, це ланка, що зв'язує наукові дослідження і практичну реалізацію. Від термінів і якості проектування в значній мірі залежать терміни впровадження і якість готової продукції. Сучасна світова тенденція націлена на значне скорочення термінів проектування.

Проектування – це і наука, і мистецтво. Однак методи проектування відрізняються від наукових методів у зв'язку з принципово різною постановкою задачі:

Шлях наукових досліджень: від явища (Е) через фактори, його зухвалі (F) до абстрактної залежності (А).

Шлях проектно-конструкторської розробки: від абстрактного завдання (А) через елементи рішення (R) до мети, що викликає бажане явище (Е).

Наукове дослідження спрямоване від явища до його сутності. Проектування – від сутності задачі до явища, що бажано викликати. Звідси головна особливість і труднощі процесу проектування: на початковому етапі мається тільки уявна, абстрактна модель майбутнього об'єкта.

Якщо виробництво виробів – є їхнє виготовлення по наявному описі, то проектування – це процес одержання такого опису.

Проектування відповідає інформаційному процесові, у якому здійснюється перетворення первинного опису (вихідного) в остаточний опис (проектну документацію) (див. рис. 1.0). При цьому використовуються знання в даній області і досвід проектувальника.

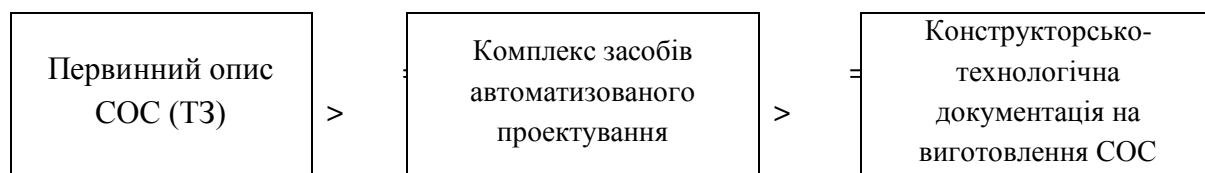


Рис. 1. Загальний інформаційний процес проектування.

Інженерне проектування - це процес, в якому наукова й технічна інформація використовується для створення нової системи, пристрою або машини, що приносять суспільству певну користь.

ГОСТ (СРСР) 22484-77 дає наступне визначення *процесу проектування*:

Проектування – це процес складання опису, необхідного (достатнього) для виготовлення в заданих умовах ще не існуючого об'єкта (алгоритму його функціонування або алгоритму процесу), шляхом перетворення первинного опису (технічного завдання), оптимізації заданих характеристик об'єкта (або алгоритму його функціонування), усунення некоректності первинного опису й послідовного подання (при необхідності) описів на різних мовах.

Проект (від латинського **projectus** - кинутий уперед) – сукупність документів і описів на

систем

різних мовах (графічній - креслення, схеми, діаграми й графіки; математичній - формули й розрахунки; інженерних термінів і понять - тексти описів, пояснювальні записки), необхідних для створення якого-небудь об'єкту або системи.

Проектування – це комплекс робіт з метою одержання опису нового або модернізованого технічного об'єкта, достатнього для реалізації або виготовлення його в конкретних умовах.

Образ об'єкта або його складових частин може створюватися в уяві людини в результаті творчого процесу або генеруватися відповідно до деяких алгоритмів у процесі взаємодії людини й ЕОМ. У будь-якому випадку інженерне проектування починається при наявності вираженої потреби суспільства в деяких технічних об'єктах, який можуть бути об'єкти будівництва, промислові виробы або процеси.

Об'єктами проектування можуть бути як виробы (електродвигуни, ЕОМ), так і процеси (технологічний, обчислювальний, управлінський), і системи (програмна, інформаційна, тощо).

В основі проектування - первинний опис - **технічне завдання** (ТЗ). Звичайно ТЗ представляють у виді набору деяких документів. Результатом проектування, як правило, служить повний комплект документації, що містить відомості достатні для виготовлення об'єкта в заданих умовах. Ця документація і є проект, точніше остаточний опис об'єкта. Більш коротко, проектування — процес, що полягає в одержанні і перетворенні вихідного опису об'єкта в остаточний опис на основі виконання комплексу робіт дослідницького, розрахункового і конструкторського характеру.

Перетворення вихідного опису в остаточне породжує ряд проміжних описів, що підводять підсумки рішення деяких задач і використовуються для обговорення і прийняття проектних рішень для закінчення або продовження проектування.

Ручне, автоматичне та автоматизоване проектування.

Проектування розділяють на неавтоматизоване (ручний режим), автоматизоване і автоматичне.

Проектування, при якому всі проектні рішення або їхня частина одержують шляхом взаємодії людини й ЕОМ, називають автоматизованим, на відміну від ручного (без використання ЕОМ) або автоматичного (без участі людини на проміжних етапах). Система, що реалізує автоматизоване проектування, являє собою систему автоматизованого проектування (САПР).

Автоматичне проектування можливе лише в окремих випадках для порівняно нескладних об'єктів. Переважаючим в даний час є автоматизоване проектування.

Проектування складних об'єктів засновано на застосуванні ідей і принципів, викладених у ряді теорій і підходів. Найбільш загальним підходом є системний підхід, ідеями якого пронизані різні методики проектування складних систем.

Ми будемо розглядати **автоматизоване проектування**, особливість якого полягає в тому, що в ході проектування відбувається постійний діалог людини з обчислювальною системою.

Призначення САПР у проектуванні СОС

Автоматизація проектування – невідемна складова сучасного науково-технічного прогресу. Проектування технічних об'єктів без автоматизації вимагає занадто великих часових і людських ресурсів. Проекти найбільш складних об'єктів, до яких відносяться інтегральні схеми (ВІС), обчислювальні системи, механічні пристрої та системи (транспортні засоби, будівельні конструкції, тощо), створюються з обов'язковим використанням САПР.

Згідно ДСТУ, який унаслідковано від ГОСТ, „САПР являє собою організаційно-технічну систему, яка складається з комплексу засобів автоматизації проектування, що взаємозв'язані з підрозділами проектної організації та виконує автоматизоване проектування”.

Отже, САПР має 2-і принципові властивості:

САПР- це організаційно-технічна система;

Функціонування САПР полягає в проектуванні певних об'єктів.

Системи автоматизованого проектування (САПР) завдяки швидкодії і надійності обчислювальної техніки, вірогідності математичних моделей (ММ) і ефективним методам оптимізації дозволяють не тільки прискорити розробку проектів, звільнити інженерів і техніків від виконання рутинних процедур, але і скоротити в цілому тривалість створення нових машин і апаратів, підвищити показники їхньої якості.

Історія та тенденції розвитку САПР

Епізодичне рішення окремих інженерних задач на ЕОМ почалось відразу після появи швидкодіючих обчислювальних машин. Перші тиражовані програми для рішення задач аналізу схем і конструювання друкованих плат з'явилися у 1-й половині 60-х років минулого століття. На початку 70-х років почали з'являтися програмні засоби у вигляді програмно-методичних комплексів (ПМК) для проектування ЕОМ і їх елементної бази, що означало появу перших САПР. В середині 70-х років промисловість почала серійно виготовляти програмно-технічні комплекси (ПТК), які в подальшому перетворились на автоматизовані робочі місця (АРМ). На початку 80-х років сформувався концепція багаторівневих САПР для наскрізного автоматизованого проектування ВІС.

Одночасно із створенням апаратних і програмних засобів проходило становлення теоретичних основ автоматизованого проектування. Важливими досягненнями стали розробка методів автоматизованого формування математичних моделей складних систем, алгоритмізація процедур проектування топології друкованих плат і ВІС, розвиток методів аналізу моделей, які представлені системами диференціальних, алгебраїчних і логічних рівнянь високих порядків, і т.п.

В наш час проводяться інтенсивні дослідження по:

Подальшій алгоритмізації процедур синтезу та аналізу об'єктів, що проектуються, які підвищують рівень інтелектуальності САПР;

Об'єднанні виробничих підсистем в єдину інтегровану систему автоматизованого проектування і виробництва деталей машинобудування та електронної техніки.

Поняття САПР та CAD/CAM/CAE/PDM

Поняття САПР є більш широким ніж CAD, оскільки включає в себе елементи CAM, CAE

САПР = CAD+CAM+CAE+PDM.

CAD - термін з'явився наприкінці 50-х років ХХ століття, коли Д.Т. Росс почав працювати над однойменним проектом у Масачусетському Технологічному Інституті (MIT) []. Однак, перші комерційні CAD системи з'явилися на ринку орієнтовно 10 років після цього проекту. Це є загальний термін, яким позначають усі аспекти інженерного проектування з використанням засобів обчислювальної техніки. Однак, найчастіше CAD системи охоплюють: системи геометричного моделювання, 2D/3D поверхневого і твердотілого моделювання, параметричного конструювання, а також системи генерації креслень і їх супровід. Незважаючи на всі свої можливості, CAD - системи надають конструкторові слабку допомогу з погляду *загального* процесу конструкторського проектування. Вони забезпечують лише опис геометричних форм і рутинні операції, такі як нанесення розмірів, генерація специфікацій і т.п. Ці обмеження і чисто геометричний інтерфейс залишає методологію конструкторської роботи такою, якою вона була при використанні креслярської дошки.

CAM - Загальний термін для позначення системи автоматизованої підготовки виробництва, загальний термін для позначення ПС підготовки інформації для верстатів з ЧПУ. Традиційно вихідними даними для таких систем були геометричні моделі деталей, отриманих з систем CAD, тобто CAM-система використовує геометричну модель CAD-системи.

CAE - Загальний термін для позначення системи автоматизованого аналізу моделі ОП, яка має на меті виявлення помилок проектування (напр., розрахунки на міцність) або оптимізацію виробничих можливостей.

PDM - Система управління виробничою інформацією. Крім проектування, інженерна діяльність зв'язана з інженерним бізнесом і менеджментом. PDM допомагає адміністраторам, інженерам, конструкторам і т.д. керувати як даними, так і процесами розробки виробів на виробничому підприємстві, або у групі суміжних підприємств.

Отже, спочатку концепція автоматизації праці конструктора базувалася на принципах геометричного моделювання і комп'ютерної графіки. При цьому, системи комп'ютеризації праці конструкторів, технологів, технологів - програмістів, інженерів - менеджерів і виробничих майстрів розвивалися автономно й Інженерні Знання - основа проектування, залишалися поза комп'ютером. Однак, така ситуація в автоматизації інженерної діяльності не задовольняла вимогам ринку. Необхідною є комплексна комп'ютеризація інженерної діяльності на всіх етапах життєвого циклу виробу, що одержала назву CALS технології (**C**omputer **A**ided **L**ife-cycle **S**ystem). Традиційні САПР із їх геометричним, а не інформаційним ядром, не можуть бути основою для створення таких систем. Сьогодні кожен виріб у процесі свого життєвого циклу повинне представлятися в комп'ютерному середовищі у виді ієрархії інформаційних моделей, що складають єдине ціле і мають таку підпорядкованість, де кожна наступна модель є більш детальною і містить додаткову інформацію про об'єкт проектування.

Традиційні CAD-CAM системи здатні допомогти інженерові лише одержати РД і УП для верстатів із ЧПУ. Для повної реалізації сучасного підходу до проектування необхідні мета-інструментальні середовища, здатні рівною мірою ефективно вирішувати як геометричні, так і негеометричні задачі.

Основні недоліки і напрямки розвитку САПР

Більшість САПР включають в себе лише засоби креслення, прискорювачі геометричних розрахунків та засоби виводу документації на друк.

рішення: запровадження комплексних наскрізних САПР, запровадження інтерактивних засобів адміністрування проектів, використання інтелектуальних інженерних БД, тощо.

- САПР не дозволяє легко і швидко вносити зміни в проект, але таким чином, щоб зміни не викликали перепроєктування уже створених деталей і вузлів. Сьогодні таке перепроєктування продовжує залишатися серйозною проблемою будь-якого проекту.

рішення: запровадження PDM - управління даними про продукцію.

- важлива проблема пов'язана з формуванням інтелектуальних функцій, тобто "розуміння" системами CAD/CAM намірів проєктувальників.

рішення: створення систем з розумінням "історії", які необхідні при створенні бібліотек стандартних деталей і елементів, та систем з розумінням "намірів", які потрібні для більш складних ситуацій, коли проект розвивається в несподіваному для проєктувальника напрямі.

- ще одна проблема - це утруднений обмін даними між САПР та іншими „зовнішніми” програмами (системами).

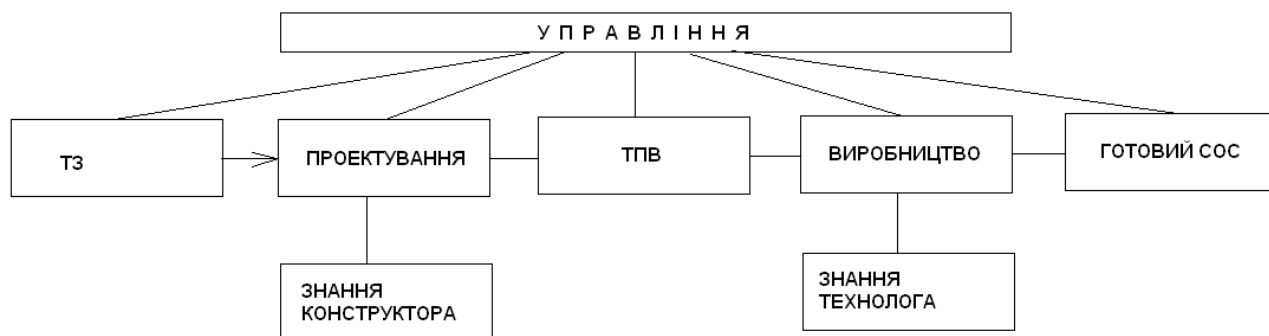
рішення: був розроблений промисловий стандарт IGES, потім він був покращений і створено стандарт STEP, але вони не підтримуються всіма розробниками ПЗ. Сьогодні великі надії покладають на стандарт XML.

- відсутність в САПР могутніх інструментальних засобів розробки, подібних CASE-технології в області СУБД, що перешкоджає адаптації і розвитку систем проектування професійним користувачем.

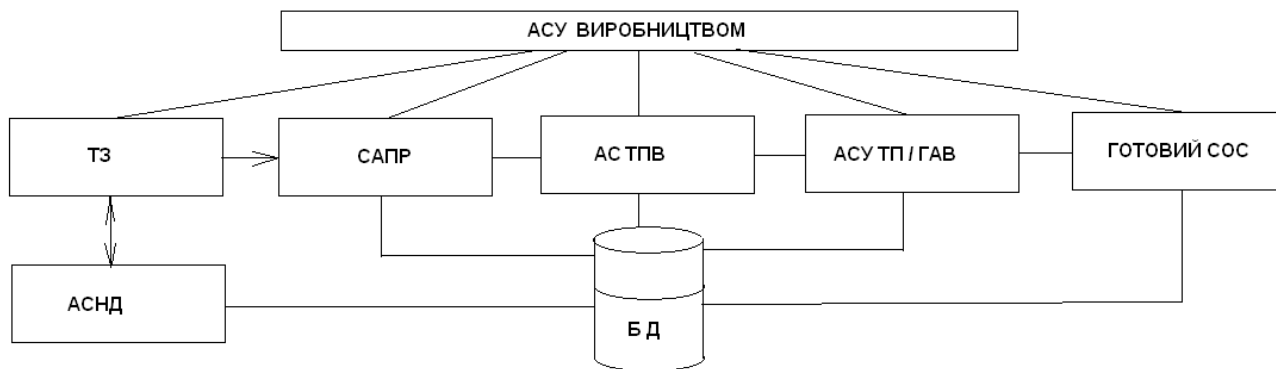
рішення: створення не окремих САПР, а інструментальних середовищ проектування, в рамках яких будуть функціонувати адаптивні САПР.

Місце САПР у сучасному виробництві

Схема проектування та виготовлення продукту на виробництві:



Відповідно до цієї схеми, функціонують та взаємодіють між собою САПР:



ТЗ - технічне завдання;

САПР - система автоматизованого проектування;

АСТПВ - автоматизована система технологічної підготовки виробництва;

АСУТП/Г АВ - автоматизована система управління технологічним процесом / гнучке автоматизоване виробництво;

АСНД - автоматизована система наукових досліджень.

Основні типи автоматизованих систем підприємства

Вимоги до якості проектів, термінів їх виконання стають все більш жорсткими у міру збільшення складності проєктованих об'єктів. Крім того, темпи морального старіння виробів сьогодні такі, що поставлені на конвеєр нові зразки часто вже не відповідають сучасним вимогам.

Досягнення поставлених цілей на сучасних підприємствах, що випускають СОС, виявляється неможливим без широкого використання *автоматизованих систем (АС)*, заснованих на застосуванні комп'ютерів і призначених для створення, переробки і використання всієї необхідної інформації про властивості виробів і супроводжуючих процесів. Специфіка завдань, що вирішуються на різних етапах життєвого циклу виробів, обумовлює різноманітність АС. На рис. вказано основні типи АС з їх прив'язкою до етапів життєвого циклу виробів.

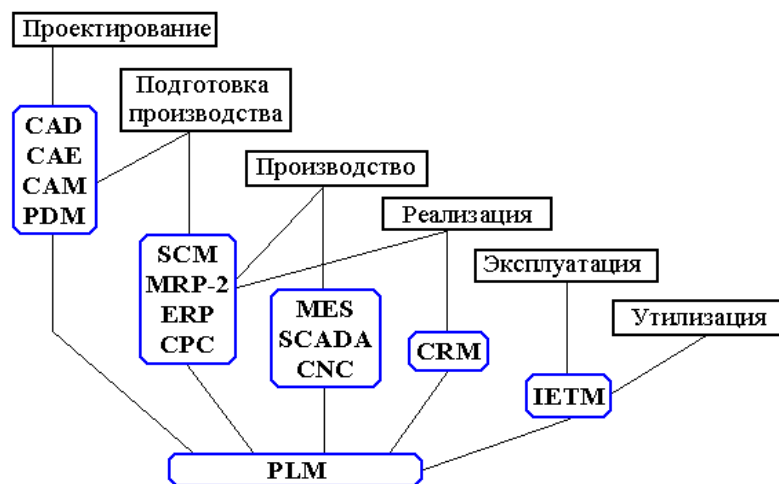


Рис. 1. Основні типи автоматизованих систем [1].

Автоматизація проектування здійснюється САПР. У САПР машинобудівних галузей промисловості прийнято виділяти системи функціонального, конструкторського і технологічного проектування.

Проектування технологічних процесів виконується в *автоматизованих системах технологічної підготовки виробництва* (АСТПП), що входять як складова частина в системи САМ.

Для вирішення проблем сумісного функціонування компонентів САПР різного призначення, координації роботи систем САЕ/CAD/CAM, управління проектними даними і процесом проектування, розробляються PDM системи. Системи PDM або входять до складу модулів конкретної САПР, або мають самостійне значення і можуть працювати спільно з різними САПР.

На більшості етапів життєвого циклу, починаючи з визначення підприємств-постачальників початкових матеріалів і компонентів і кінчаючи реалізацією продукції, потрібні послуги логістичної системи - Supply Chain Management (SCM). Ланцюг постачань зазвичай визначають як сукупність стадій збільшення доданої вартості продукції при її русі від компаній-постачальників до компаній-споживачів. Управління ланцюгом постачань має на меті формування матеріального потоку з мінімальними витратами. При плануванні виробництва система SCM управляє стратегією позиціонування продукції. Якщо час виробничого циклу менше часу очікування замовника на отримання готової продукції, то можна застосовувати стратегію "виготовлення на замовлення". Інакше доводиться використовувати стратегію "виготовлення на склад". При цьому під час виробничого циклу повинен входити час на розміщення і виконання замовлень на необхідні матеріали і що комплектують на підприємствах-постачальниках.

Управління в промисловості, як і в будь-яких, має ієрархічну структуру. У загальній структурі управління виділяють декілька ієрархічних рівнів, показаних на рис. 2. Автоматизація управління на різних рівнях реалізується з допомогою *автоматизованої системи управління* (АСУ).



Рис. 2. Загальна структура АСУ

Інформаційна підтримка етапу виробництва продукції здійснюється

систем

автоматизированными системами управления предприятием (АСУП) і автоматизированными системами управления технологическими процессами (АСУТП).

До АСУП відносяться системи планування і управління підприємством ERP (Enterprise Resource Planning), планування виробництва і вимог до матеріалів MRP-2 (Manufacturing Requirement Planning) і згадані вище системи SCM. Найбільш розвинені системи ERP виконують різні бизнес-функції, пов'язані з плануванням виробництва, закупівлями, збутом продукції, аналізом перспектив маркетингу, управлінням фінансами, персоналом, складським господарством, обліком основних фондів і тому подібне. Системи MRP-2 орієнтовані, головним чином, на бизнес-функції, безпосередньо пов'язані з виробництвом. В деяких випадках системи SCM і MRP-2 входять як підсистеми в ERP, останнім часом їх гушавині розглядають як самостійні системи.

Проміжне положення між АСУП і АСУТП займає виробнича виконавська система MES (Manufacturing Execution Systems), призначена для вирішення оперативних завдань управління проектуванням, виробництвом і маркетингом.

До складу АСУТП входить система SCADA (Supervisory Control and Data Acquisition), що виконує диспетчерські функції (збір і обробка даних про стан устаткування і технологічних процесів) і що допомагає розробляти ПО для вбудованого устаткування. Для безпосереднього програмного управління технологічним устаткуванням використовують системи CNC (Computer Numerical Control) на базі *контролерів* (спеціалізованих комп'ютерів, званих промисловими), які вбудовані в технологічне устаткування з числовим програмним управлінням (ЧПУ).

На етапі реалізації продукції виконуються функції управління відносинами із замовниками і покупцями, проводиться аналіз ринкової ситуації, визначаються перспективи попиту на плановані вироби. Ці функції покладені на систему CRM.

Управління даними в єдиному інформаційному просторі впродовж всіх етапів життєвого циклу виробів покладається на систему управління життєвим циклом продукції PLM (Product Lifecycle Management). Характерна особливість PLM — забезпечення взаємодії різних автоматизованих систем багатьох підприємств, тобто технології PLM (включаючи технології CPC) є основою, інтегруючою інформаційний простір, в якому функціонують САПР, ERP, PDM, SCM, CRM і інші автоматизовані системи багатьох підприємств.

Життєвий цикл виробів

Будь-який технічний об'єкт чи система, що створюються людиною, в кінці кінців, зношується в процесі експлуатації або застаріває морально, тобто перестає задовольняти збільшеним вимогам.

Інтервал часу від початку створення технічного об'єкта до кінця його експлуатації, називають "*життєвим циклом*" технічного об'єкта. При цьому, за початок життєвого циклу традиційно прийнято вважати зародження ідеї про необхідність створення системи (визначення концепції), а за кінець - зняття її з експлуатації (утилізацію).

Життєвий цикл виробу включає ряд етапів, починаючи від зародження ідеї нового продукту до його утилізації після закінчення терміну використання. Основні етапи життєвого

циклу промислової продукції представлені на рис. [1] До них відносяться як підготовчі етапи, так і етапи проектування, виготовлення і експлуатації.

На етапі проектування виконуються проектні процедури — формування принципового рішення, розробка геометричних моделей і креслень, розрахунки, моделювання процесів, оптимізація тощо.

На етапі підготовки виробництва розробляються маршрутна і операційна технології виготовлення деталей, що реалізуються в програмах для верстатів ЧПУ; технологія збірки і монтажу виробів; технологія контролю і випробувань.

На етапі виробництва здійснюються: календарне і оперативне планування; придбання матеріалів і що комплектують з їх вхідним контролем; механообробки і інші необхідні види обробки; контроль результатів обробки; збірка; випробування і підсумковий контроль.

На пост-виробничих етапах виконуються консервація, упаковка, транспортування; монтаж у споживача; експлуатація, обслуговування, ремонт; утилізація.

Рис. 2 Структура життєвого циклу технічного об'єкта.

На всіх етапах життєвого циклу є свої цільові установки. При цьому учасники життєвого циклу прагнуть досягти поставлених цілей з максимальною ефективністю. На етапах проектування, ТПП і виробництва потрібно забезпечити виконання вимог, що пред'являються до вироблюваного продукту, при заданому ступені надійності виробу і мінімізації матеріальних і часових витрат, що необхідне для досягнення успіху в конкурентній боротьбі в умовах ринкової економіки. Поняття ефективності охоплює не тільки зниження собівартості продукції і скорочення термінів проектування і виробництва, але і забезпечення зручності освоєння і зниження витрат на майбутню експлуатацію виробів. Особливу важливість вимоги зручності експлуатації мають для складної техніки, наприклад, в таких галузях, як авіа- або автомобілебудування.

Основними стадіями життєвого циклу будь-якого технічного об'єкта є наступні [2]:

- 1) маркетингові дослідження потреб ринку;
- 2) генерація ідей та їх фільтрація;
- 3) технічна та економічна експертиза проекту;
- 4) науково-дослідні роботи за тематикою виробу;
- 5) дослідно-конструкторська робота;
- 6) виробництво дослідного зразка, пробний маркетинг;
- 7) підготовка та виробництво виробу на заводі-виробнику серійної продукції;
- 8) маркетинг і збут виробів;
- 9) експлуатація, підтримка і супровід виробів;
- 10) утилізація виробів.

Стадії з 4-ї по 7-у являються перед-виробничі, і їх можна розглядати як комплекс науково-технічної підготовки виробництва [2]. Основні параметри, що характеризують кордону стадій життєвого циклу виробу, наведені в табл. 1.

Межі стадій життєвого циклу виробу

Стадія	Початок стадії	Закінчення стадії
Маркетингові дослідження ринку	Укладання договору на проведення досліджень	Здача звіту за результатами досліджень
Генерація ідей і їх фільтрація	Збір і фіксація пропозицій щодо проектів	Закінчення відбору проектів-конкурентів
Технічна і економічна експертиза проектів	Комплектація груп оцінки проектів	Здача звіту з експертизи проектів, вибір проекту-переможця
НДР	Затвердження ТЗ на НДР	Затвердження акту про закінчення НДР
ДКР	Затвердження ТЗ на ДКР	Наявність комплексу конструкторської документації, відкоригованої за результатами випробувань дослідного зразка
Пробний маркетинг	Початок підготовки виробництва дослідної партії	Аналіз звіту про результати пробного маркетингу
Підготовка виробництва на заводі-виробнику	Прийняття рішення про серійне виробництво і комерційної реалізації виробів	Початок усталеного серійного виробництва
Власне виробництво і збут	Продаж першого серійного зразка виробу	Постачання споживачеві останнього екземпляра виробу
Експлуатація	Отримання споживачем першого примірника виробу	Зняття з експлуатації останнього екземпляра виробу
Утилізація	Момент списання першого примірника виробу з експлуатації	Завершення робіт з утилізації останнього виробу, знятого з експлуатації

Таким чином, можна зробити наступний висновок про те, що основним змістом цільових досліджень у процесі управління життєвим циклом виробу є: аналіз прогнозованого стану об'єктів, визначення очікуваних і фактичних результатів, оцінка пріоритетності у вирішенні локальних задач, виявлення кращих напрямів використання ресурсів.

Як вже зазначалося вище, при такому аналізі виникають наступні питання:

- - Які фактори, умови і на яких стадіях слід піддавати оцінці?
- - Якою повинна бути система критеріїв оцінок?
- - Які методологічні підходи та прийоми слід використовувати в ході оцінки?

Доцільно в ході управління життєвим циклом виробу спиратися на систему контрольних точок циклу. На всіх контрольних точках аналізують відхилення якісних та кількісних параметрів виробу від проектних значень за технічними та економічними критеріями і

виробляють відповідні рішення за критерієм «ефект-витрати».

Кількість контрольних точок (КТ) залежить від характеру виробу. Ми вважаємо, що можна виділити наступні контрольні точки (КТ) у структурі життєвого циклу технічного об'єкта:

КТ-1 - рішення про початок проекту;

КТ-2 - закінчення технічного проекту (рішення про розробку робочої документації та виготовлення дослідного зразка);

КТ-3 - закінчення ДКР (рішення про виготовлення дослідного зразка);

КТ-4 - закінчення пробного маркетингу (прийняття рішення про початок серійного виробництва і комерційної реалізації виробу);

КТ-5 - оцінка якості серійно випускається (рішення про підвищення якості та надійності);

КТ-6 - оцінка необхідності оновлення або модернізації продукції;

КТ-7 - оцінка оптимальності методів збуту продукції;

КТ-8 - оцінка доцільності та методів капітального ремонту виробів в процесі експлуатації;

КТ-9 - оцінка доцільності зняття виробу з виробництва;

КТ-10 - зняття виробу з експлуатації і передача його на утилізацію.

Тривалості всіх стадій життєвого циклу виробу корінним чином впливають на його економічну ефективність [3].

Особливе значення має скорочення термінів науково-технічної підготовки виробництва, в тому числі і забезпечення певної паралельності виконання окремих етапів. Для цього необхідно:

Знизити до мінімуму всі зміни, що вносяться до виріб після передачі результатів від одного етапу до іншого;

Визначити і реалізувати раціональну паралельність робіт, фаз, стадій циклу;

Забезпечити скорочення витрат часу на виконання окремих етапів.

Вирішення першого завдання забезпечується інженерно-технічними методами (стандартизація, уніфікація, забезпечення якості та надійності, застосування САПР і т. д.).

Вирішення другого завдання здійснюється шляхом застосування планово-координаційних методів.

Рішення третьої задачі пов'язано з першою і полягає у використанні організаційних методів (розвиток технічного забезпечення, автоматизації, засобів планування, функціонально-вартісного аналізу, дослідного виробництва).

СТАДІЯ ПРОЕКТУВАННЯ

Проектування - це комплекс робіт з метою отримання описів нового або модернізованого технічного об'єкта, достатніх для реалізації або виготовлення об'єкта в заданих умовах [1]. Об'єктами проектування можуть бути вироби (наприклад, ЛА, ЕОМ) або процеси (наприклад, процес управління автоматичним КА).

Комплекс проектних робіт включає в себе теоретичні та експериментальні дослідження, конструювання, розробку технічної та проектної документації [1].

О. В Афанасьєва в процесі проходження стадії проектування виділяє ще дві стадії:

систем

концептуального і технічного проектування [4]. Вона вважає, що на стадії концептуального проектування визначається необхідність і принципова можливість (здійсненність створення конкретної системи); виробляються цілі та критерії її застосування та проектування, визначається зовнішній вигляд системи; обґрунтовуються основні тактико - технічні характеристики і оцінюються ресурси, необхідні для подальших робіт; формалізується і узгоджується з Виконавцем завдання на технічне проектування системи [1, с. 5].

Стадія ж технічного проектування, на думку О. В. Афанасьєвої включає в себе такі етапи, як технічне завдання, технічна пропозиція, аванпроект, ескізний проект, технічний проект і робочий проект [1, с. 5]. Найдовшим і трудомістким етапом тут є робочий проект, в процесі якого виготовляються дослідні зразки та проводяться випробування з подальшим коректуванням за їх результатами технічної документації. Цей цикл (доведення виробу) повторюється до тих пір, поки дослідний зразок не буде повністю задовольняти вимоги технічного завдання (ТЗ).

Вона звертає увагу, тим не менш, на ту обставину, що традиційно етапи технічного і робочого проектування на практиці прийнято об'єднувати під загальною назвою дослідно - конструкторські роботи (ДКР), які традиційно закінчуються випробуваннями зразка системи. Саме за результатами цих випробування і приймається рішення про прийняття технічного об'єкта і його виробництві [1, с. 6].

Ще більш докладний поділ стадії проектування міститься у роботі А. А. Лебедева [6].

Він вважає, що спочатку слід виділити перший етап, який він визначає як зовнішнє проектування [6, с. 56]. Таке найменування, на його думку, пов'язано з тим обставиною, що проєктований технічний об'єкт розглядається як компонент більш грубих технічних системи (над-системи), враховується взаємодія технічного об'єкта з навколишнім середовищем, і, зокрема, з іншими компонентами над-системи, оцінюються результати, які отримає над-система від створення проєктованого технічного об'єкта, і витрати ресурсів на її створення. Цей етап звичайно виконується спільними зусиллями замовника та розробників в НДІ замовника технічного об'єкта, а також в НДІ і ОКБ розробників. У рамках реалізації даного етапу відбувається формулювання загальних вимог до проєктованого технічного об'єкта, які будуть служити вихідними даними для подальших дослідно - конструкторських робіт.

Зовнішнє проектування технічного об'єкта починається з встановлення необхідності створення нового технічного об'єкта і формулювання його призначення і завдань. Потім визначаються основні обмеження на технічне, економічні та організаційні рішення, в рамках яких повинен надалі розроблятися проєктований технічний об'єкт; прикладами таких обмежень можуть стати тип старту летального апарату, або вид палива для рухової установки, терміни створення та експлуатації технічного об'єкта, припустимі витрати на його створення. Результатом зовнішнього проектування є створення тактико - технічного завдання для подальшого проведення дослідно - конструкторських робіт. Дані результати зовнішнього проектування дають можливість в подальшому приступити до реалізації наступного етапу стадії проектування.

У рамках другого етапу стадії проектування, на думку Лебедева, здійснюються дослідно - конструкторські роботи, метою проведення яких є реалізації тактико-технічного завдання.

На під-етапі технічної пропозиції роботи розгортаються в умовах, коли на проєктований технічний об'єкт вже накладені деякі вимоги і обмеження, які були визначені на етапі зовнішнього проєктування.

У свою чергу одним із завдань технічної пропозиції є формування вимог і обмежень, в рамках яких буде відбуватися проєктування самого технічного об'єкта. Для цього уточнюється вигляд технічного об'єкта, тобто коригується тактико-технічне завдання, і типи елементів, встановлюються діапазони характеристик елементів, які забезпечують виконання технічним об'єктом свого прямого призначення. Підсумком робіт на даному етапі є сформульоване технічну пропозицію, тобто комплекс документації, що включає в себе пояснювальну записку в декількох томах, альбом креслень та науково - технічні розрахунки. У ньому міститиметься прийняті рішення на технічний об'єкт, технічні завдання на проєктування елементів, перелік робіт, які необхідно зробити в наступному ескізному проєктуванні. На підставі технічної пропозиції приймається рішення про подальше проведення ескізного та робочого проєктування.

На під-етапі ескізного проєкту перевіряються, конкретизуються принципи та положення технічної пропозиції, приймаються докладні технічні рішення та опрацьовуються всі частини проєкту. Підсумком робіт на даному підетапі є ескізний проєкт технічного об'єкта.

На під-етапі робочого проєктування створюється повний комплект проєктно - конструкторської, технологічної та експлуатаційної документації, достатній для виготовлення елементів технічного об'єкта.

На під-етапі виготовлення і обробки дослідних зразків і подальших їх випробувань виявляються можливі помилки або недоробки проєкту, вживаються заходи щодо їх усунення.

Підсумками робіт на даних етапів є рішення про серійне виробництво технічного об'єкта.

Таким чином, необхідно зробити ряд наступних висновків.

Проєктування - це комплекс робіт з метою отримання описів нового або модернізованого технічного об'єкта, достатніх для реалізації або виготовлення об'єкта в заданих умовах [6, с. 59].

У рамках розгляду концепції життєвого циклу технічного об'єкта проєктування являє собою стадію від формулювання ідеї технічного об'єкта до початку його серійного виробництва. Ця стадія є витратною, надзвичайно складною і трудомісткою.

СТАДІЯ ВИРОБНИЦТВА

На стадії виробництва проводиться технологічна підготовка виробництва, виготовлення, збірка, настройка, заводські випробування і складування готової продукції [1, с. 5].

Головне завдання підготовки виробництва - створення та організація випуску нових виробів [8, с. 24].

Організація процесів створення нових видів продукції машинобудування охоплює проєктування, здійснення на практиці і вдосконалення системи підготовки виробництва.

Система підготовки виробництва - це об'єктивно існуючий комплекс матеріальних об'єктів, колективів людей і сукупність процесів наукового, технічного, виробничого та економічного характеру для розробки та організації випуску нової або вдосконаленої продукції [8, с. 24].

Організація підготовки виробництва виражається наступних видах діяльності:

- Визначення мети організації та її орієнтація на досягнення цієї мети;
- Встановлення переліку всіх робіт, які повинні бути виконані для досягнення поставленої мети щодо створення конкретних видів нової продукції;
- Створення або вдосконалення організаційної структури системи підготовки виробництва на підприємстві;
- Закріплення кожної роботи за відповідним підрозділом (відділом, групою, цехом і т. п.) підприємства;
- Організація робіт по створенню нових видів продукції в часі;
- Забезпечення раціональної організації праці працівників та необхідних умов для здійснення) всього комплексу робіт з підготовки виробництва до випуску нової продукції;
- Встановлення економічних відносин між учасниками процесу створення нової техніки.

Організація підготовки виробництва будується на реалізації наступних принципів:

1. Принцип комплексності передбачає необхідність проведення робіт з підготовки виробництва за єдиним планом, який охоплює всі процеси - від наукових досліджень до освоєння нової техніки - враховує виникають при цьому.

2. Принцип спеціалізації вимагає, щоб за кожним підрозділом підприємства закріпилися такі види діяльності щодо створення та освоєння нової техніки, які відповідають характеру спеціалізації цих підрозділів.

3. Принцип науково-технічної і виробничо інтеграції - це сукупність умов, що забезпечують досягнення єдиної і спільної мети в результаті діяльності певної множини спеціалізованих підрозділів і виконавців.

4. Принцип комплектності документації та складових частин виробів вимагає одночасного виконання комплексу робіт до моменту, коли подальше їх продовження можливе тільки за наявності повного комплексу документації або складових частин виробів.

5. Принцип безперервності робіт зі створення нової продукції вимагає ліквідації значних перерв у часі між фазами процесу підготовки, а всередині їх між стадіями, роботами, операціями.

6. Принцип пропорційності розглядається як вимога задіяти виробничі можливості всіх підрозділів об'єднання або підприємств, зайнятих підготовкою виробництва.

7. Принцип паралельності виражається в поєднанні в часі різних фаз, стадій, робіт.

Система підготовки виробництва передбачає забезпечення суворої послідовності робіт і прямо точності. Цей принцип вимагає, щоб розробка та освоєння нової продукції здійснювалися з притаманною тільки цього виду послідовністю робіт. Прямо точність - це найкоротший маршрут руху технічної документації і найменший шлях, прохідний новим виробом по всіх стадіях його розробки та освоєння.

Перехід підприємств на випуск нового виробу може виконуватися такими методами: послідовним, паралельним, комплексно-поєднаним і агрегатним.

Послідовним називається такий перехід, коли виробниче освоєння починається тільки

після зняття з виробництва раніше випускався вироби. Технічна та організаційна підготовка виконується заздалегідь - під час випуску старої продукції.

Паралельний метод переходу передбачає максимальне суміщення виробництва знову освоюваних виробів із завершальною стадією випуску старої моделі. Він зазвичай застосовується при наявності у підприємства резервних потужностей, створення паралельно діючих ділянок, конвеєрів. При послідовному і паралельному методах освоюється випуск всієї нової машини в цілому при повній готовності до виробництва всіх вузлів.

Комплексно-суміщений метод характеризується поєднанням виконання окремих робіт з підготовки виробництва до освоєння нових виробів при комплексному вирішенні конструкторських, технологічних і виробничих завдань. Виробничники беруть участь у проектуванні виробу, розробники - в освоєнні його випуску. Цей метод дозволяє значно прискорити процес створення нової продукції за рахунок скорочення процедури оформлення та затвердження технічної документації, виключення зайвих робіт, виконання блокового проектування і виготовлення різних вузлів, часткового суміщення різних робіт, виконання переходу до серійного виробництва без виготовлення дослідних зразків і дослідно-промислових партій.

Агрегатний метод передбачає поступову заміну окремих агрегатів в конструкції випускається старої моделі. Протягом деякого часу випускається перехідний модифікований виріб, забезпечене тільки окремими новими вузлами. При завершенні запланованої заміни старих агрегатів новими модель з перехідною перетворюється на новий виріб. Освоєння ділиться на кілька етапів, колектив зосереджує зусилля на порівняно невеликій ділянці роботи, і перехід відбувається менш болісно для підприємства. При виборі методу переходу слід враховувати фактори, що характеризують організаційно-технічний рівень виробництва, конструкцію нового виробу і технологію виробництва.

Оцінюючи організаційно-технічні умови, необхідно враховувати:

- - Наявність резерву виробничих потужностей;
- - Наявність вільних виробничих площ;
- - Внутрішньозаводських спеціалізацію і розподіл праці в цехах і на дільницях;
- - Рівень галузевої та міжгалузевої кооперації;
- - Наявність кваліфікованих кадрів.

Особливе значення в рамках даної стадії грає також форма організації виробництва.

Форма організації виробництва являє собою певне поєднання в часі і просторі елементів виробничого процесу при відповідному рівні його інтеграції, виражене системою стійких зв'язків.

Різні структурні побудови в часі і в просторі утворюють сукупність основних форм організації виробництва.

Часова структура форм організації виробництва визначається складом елементів виробничого процесу та порядком їх взаємодії в часі.

За виду часової структури розрізняють форми організації з послідовною, паралельною і паралельно-послідовною передачею предметів праці у виробництві.

Форма організації з послідовною передачею предметів праці є таке поєднання елементів

систем

виробничого процесу, при якому забезпечується рух оброблюваних виробів по всіх виробничих ділянках партіями довільної величини. Предмети праці на кожну наступну операцію віддаються лише після закінчення переробки всієї партії на попередній операції. Дана форма є найбільш гнучкою по відношенню до змін, що виникають у виробничій програмі, дозволяє досить повно використовувати обладнання, що дає можливість знизити значні витрати на його придбання.

Недолік форми з послідовною передачею предметів праці полягає у відносно великій тривалості циклу, так як кожна деталь перед виконанням наступної операції пролежівает в очікуванні обробки своєї партії.

Форма організації з паралельною передачею предметів праці заснована на такому поєднанні елементів виробничого процесу, що дозволяє запускати, обробляти і передавати предмети праці з операції на операцію поштучно і без очікування. Така організація виробничого процесу призводить до зменшення кількості деталей, що знаходяться в обробці, скорочення потреби в площах, необхідних для складування і проходів. Недолік форми полягає в можливих простоях обладнання (робочих місць), що виникають внаслідок різниці в тривалості операцій.

Форма організації з паралельно-послідовною передачею праці є проміжною між послідовної і паралельної формами і частково усуває властиві їм недоліки. Вироби з операції на операцію передаються транспортними партіями. При цьому забезпечується безперервність використання обладнання і робочої сили, часткове паралельне проходження партії деталей по операціях технологічного процесу.

Просторова структура форм організації виробництва визначається кількістю технологічного обладнання, зосередженого на робочому майданчику (число робочих місць), і розташуванням їх відносно напрямку руху предметів праці і навколишнього простору.

У залежності від кількості технологічного обладнання (робочих місць) розрізняють одночасну виробничу систему і відповідну їй структуру відокремленого робочого місця і багатоланкову систему з цехової, лінійної або комірчастої структурою.

Цехова просторова структура характеризується створенням ділянок, на яких обладнання (робоче місце) розташоване паралельно потоку заготовок, що передбачає їх спеціалізацію за ознакою технологічної однорідності. У цьому випадку партія деталей, які надходять на ділянку, спрямовується на одне з вільних робочих місць, де відбувається відповідний цикл обробки, після чого передається на іншу ділянку (в цех).

На ділянці з лінійної просторової структурою робочі місця (устаткування) розташовуються по ходу технологічного процесу, і партія деталей, що обробляється на ділянці, передається з одного робочого місця на інше в прямій послідовності.

Комірчаста просторова структура об'єднує ознаки лінійної і цехової. Комбінація просторової і часової структури виробничого процесу при певному рівні інтеграції часткових процесів обумовлює різні форми організації виробництва: технологічну, предметну, прямоточну, інтегровану.

Технологічна форма організації виробництва характеризується цеховою структурою і послідовною передачею предметів праці.

Предметна форма організації виробництва має пористу структуру з паралельно-послідовною (послідовної) передачею предметів праці у виробництві. На предметному ділянці встановлюється, як правило, все обладнання, необхідне для обробки групи деталей з початку і до кінця технологічного процесу. Якщо технологічний цикл замикається в межах ділянки, він називається предметно-замкнутим.

Прямоточна форма організації виробництва характеризується лінійною структурою з поштучною передачею предметів праці. Така форма забезпечує реалізацію низки принципів організації - прямоточності, безперервності, паралельності циклу, більш ефективного використання робочої сили за рахунок великої спеціалізації праці, зменшення обсягу незавершеного виробництва.

При точковій форми організації виробництва робота повністю виконується на основному робочому місці. Вироби виготовляються там, де знаходяться його основні частини. Як приклад може служити збірка вироби з переміщенням робочого навколо нього.

Інтегрована форма організації виробництва передбачає об'єднання основних і допоміжних операцій в єдиний інтегрований процес з комірчастою та лінійною структурою при паралельній, паралельно-послідовною, послідовної передачі предметів праці і виробництва.

Залежно від здатності до переналагодженні на випуск нових виробів перераховані вище форми організації умовно можна розділити на гнучкі (переналагоджувані) і жорсткі (непереналаживаеміе). Жорсткі форми організації виробництва припускають обробку деталей одного найменування. Зміна в номенклатурі продукції, що випускається і перехід на випуск конструктивно нової серії виробів викликає необхідність перепланування ділянки, заміни обладнання та оснащення. До числа жорстких відноситься потокова лінія організації. Гнучкі форми організації виробництва дозволяють забезпечити перехід на випуск нових виробів без зміни складу компонентів виробничого процесу при незначних витратах праці і часу.

Найбільшого поширення на машинобудівних підприємствах в даний час отримали такі форми організації виробництва, як гнучке точкове виробництво, гнучка предметна точкова форма.

Гнучке точкове виробництво передбачає просторову структуру відокремленого робочого місця без подальшої передачі предметів праці в процесі виробництва. Деталі повністю обробляються на одній позиції. Пристосованість до випуску нових виробів здійснюються за рахунок зміни робочого місця системи.

Гнучка предметна форма організації виробництва характеризується можливістю автоматичної обробки деталей у межах певної номенклатури без переривання на переналагодження. Перехід до випуску нових виробів здійснюється шляхом переналажуванні технічних засобів. Гнучка предметна форма охоплює область послідовної й паралельно-послідовної передачі предметів праці в поєднанні з комбінованою просторовою структурою.

Гнучка прямолінійна форма організації виробництва характеризується швидкою переналадкою на обробку нових виробів у межах заданої номенклатури шляхом заміни інструментального оснащення і пристосувань, перепрограмування системи управління. Вона заснована на рядном розташуванні устаткування, суворо відповідному технологічним процесом з поштучною передачею предметів праці.

Під впливом науково-технічного прогресу в техніці і в технології машинобудування відбуваються суттєві зміни, зумовлені механізацією і автоматизацією виробничих процесів. Це

систем

створює об'єктивні передумови розвитку нових форм в організації виробництва. Однією з таких форм, що отримала застосування при впровадженні засобів гнучкої автоматизації в виробничий процес, є блочно-модульна форма організації виробництва.

Створення виробництва блочно-модульної форми здійснюється шляхом концентрації на ділянці всього комплексу обладнання, необхідного для безперервного виробництва обмеженої номенклатури виробів, і об'єднання групи робітників на випуску кінцевої продукції при передачі їм частини функцій з планування виробництва на ділянці.

Ці стадії включають заходи з організації виробництва нового виробу або освоєного іншими підприємствами.

Вихід на потужність відбудеться після завершення робіт з підготовки виробництва: пуск і перевірка технологічного обладнання; запуск у виробництво установчої серії; проведення кваліфікаційних випробувань виробів установчої серії; доопрацювання та коригування технологічної та іншої документації.

Установча серія або перша промислова партія виробів виготовляється для перевірки здатності даного виробництва забезпечити промисловий випуск продукції відповідно до вимог науково-технічної документації (НТД) і споживачів. Зразки настановної партії, що пройшли прийнятно-здавальні та кваліфікаційні випробування, можуть бути представлені на ринку нововведень (проведення рекламної кампанії, демонстрація на виставках, торгових центрах тощо).

СТАДІЯ ЕКСПЛУАТАЦІЇ

Всі розглянуті стадії життєвого циклу (НДР, ДКР, ПП і ВМ) носять назву перед-виробничі. Тут формується виріб, його якість; закладається технічний рівень виробу, його прогресивність.

Наступною стадією життєвого циклу є виробництво виробу у відповідності зі сформованим портфелем замовлень.

Завершальною стадією життєвого циклу є експлуатація (для виробів тривалого користування) або споживання (для сировини, палива тощо) замовником чи споживачем, які використовують дану продукцію за призначенням або як комплектуючі вироби при виробництві іншої продукції.

Взаємовідносини між споживачем і виробником продукції визначається договором на поставку. Важливо забезпечити систематичне оновлення продукції за рахунок випуску нових виробів і зняття з виробництва застарілих.

Тривалість життєвого циклу в кожен конкретний період науково-технічного прогресу визначається фізичним і моральним терміном старіння техніки незалежно від термінів виконання та організації робіт за стадіями життєвого циклу і всередині них по етапах.

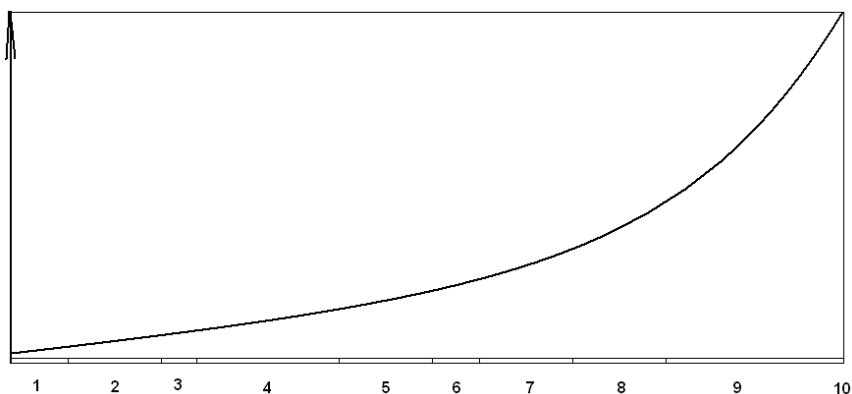
Фактори, що впливають на показники надійності системи протягом її життєвого циклу, закладаються на етапі проектування. При цьому необхідно так збалансувати витрати на розробку та проектування системи з витратами на її експлуатацію, щоб загальна сума витрат не перевищувала задану при забезпеченні найкращих характеристик системи.

Задачі, що виникають на етапі проектування, докладно описані у відповідній літературі

[2, с.9]. У загальному випадку завдання з області надійності є задачами оптимізації. Точність вирішення таких завдань залежить від точності оцінюваних даних, в тому числі і від показників надійності. Життєдіяльність будь-якого технічного об'єкта чи системи протікає всередині організаційних систем, причому окремі її стадії і навіть етапи, часом забезпечуються різними організаціями.

ПОМИЛКИ ПРОЕКТУВАННЯ

Як показано на схемі нижче, якщо життєвого циклу в конструкції об'єкта виявляють помилки, то вони ведуть до певних затрат на їх усунення. Величина цих затрат дуже сильно залежить від поточної стадії життєвого циклу, коли їх було виявлено:



Висновки

Таким чином, підводячи підсумок всьому вищесказаному, необхідно зробити ряд наступних висновків.

Будь-який технічний об'єкт, створений людиною, в кінці кінців, зношується в процесі експлуатації або застаріває морально, тобто перестає задовольняти збільшеним вимогам.

Інтервал часу від початку створення технічного об'єкта до кінця його експлуатації, називають життєвим циклом технічного об'єкта, при цьому за початок життєвого циклу традиційно прийнято розуміти зародження ідеї про необхідність створення системи, а за кінець - зняття її з експлуатації [6, с. 55].

Життєвий цикл будь-якого технічного об'єкта можна розділити на три етапи: етап проектування, етап виробництва і етап експлуатації.

Кожна з цих стадій, у свою чергу складається з окремих етапів. Етапи технічного проектування і зміст виконуваних на них робіт суворо визначені ГОСТом, чого не можна сказати про інші стадії.

Фактори, що впливають на показники надійності системи протягом її життєвого циклу, закладаються на етапі проектування.

При цьому необхідно так збалансувати витрати на розробку та проектування системи з витратами на її експлуатацію, щоб загальна сума витрат не перевищувала задану при забезпеченні найкращих характеристик системи.

Задачі, що виникають на етапі проектування, докладно описані у відповідній літературі [2, с.9].

У загальному випадку завдання з області надійності є задачами оптимізації. Точність вирішення таких завдань залежить від точності оцінюваних даних, в тому числі і від

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем показників надійності.

Життєдіяльність будь-якого технічного об'єкта чи системи протікає всередині організаційних систем, причому окремі її стадії і навіть етапи, часом забезпечуються різними організаціями.

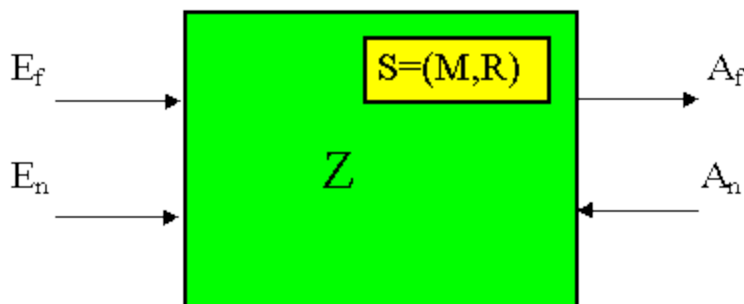
Література

1. И.П.Норенков, В.Б.Маничев Основы теории и проектирования САПР. М.: Высшая школа, 1990. -335 с.
2. К.Зиглер. Методы проектирования программных систем. М.: Мир, 1985. -328 с.
3. Системы автоматизированного проектирования (в 9 кн.). Под ред И.П.Норенкова. М.: Радио и связь, 1986. -368 с.
4. Корячко В.П. и др. Теоретические основы САПР. М.:Энергоатомиздат, 1987. -335с.
5. Построение экспертных систем. Под ред. Ф.Хейеса-Рата. М.: Мир, 1987. -441 с.
6. Системы автоматизированного проектирования в радиоэлектронике. Справочник. Под ред И.П.Норенкова. М.:Радио и связь, 1986. -368 с.
7. Волкович В.Л. и др. Методы и алгоритмы автоматизированного проектирования сложных систем управления.-Киев:Наукова думка, 1984. -216с.

2. Поняття складного об'єкта чи системи

Технічна система та її елементи.

Технічна система (з теорії технічних систем) – це обмежена область реальної дійсності, що взаємодіє з навколишнім середовищем U , яке виконує певні функції F і має структуру S .



E_f, A_f - параметри, що характеризують функції F системи;

E_n, A_n - параметри, що не ставляться до функцій приладу (умови роботи, зовнішні й додаткові впливи);

Z - системний оператор;

M - елементи системи;

R - відносини між елементами системи.

Навколишнє середовище U - сукупність зовнішніх об'єктів, взаємодіючих із системою.

Функція F - властивість системи, використовувана для перетворення вхідних величин E_f , при зовнішніх і додаткових впливах A_n і умовах роботи E_n , у вихідні величини A_f . Функція є об'єктивно вимірювана властивість, що може бути охарактеризовано параметрами системи. Кількість реалізованих системою функцій відповідає кількості використовуваних системою фізичних властивостей. Якщо система виконує кілька функцій, то розрізняють загальну й часткові функції. Загальна функція охоплює множину всіх вхідних і вихідних величин, що характеризує розглянуту систему як одне ціле. Часткові функції діляться на: головні й допоміжні - по їхньому значенню у виконанні завдання; основні й елементарні - по типі зміни змін функцій у процесі їхнього виконання.

Структура S - сукупність елементів M і відносин R між ними усередині системи $S=(M,R)$. Елемент системи при проектуванні розглядається, як одне ціле, хоча він може мати різний ступінь складності. Якщо при розгляді елемента, не приймається в увагу його форма й внутрішня будова, а розглядається тільки виконувана їм функція, то такий елемент називається функціональним. Для механічної системи елементами можуть бути: деталь, ланка, група, вузол, простий або типовий механізм.

Деталь - елемент конструкції не має у своєму складі внутрішніх зв'язків (що складає з одного твердого тіла).

Ланка - тверде тіло або система жорстко зв'язаних твердих тіл (може складатися з однієї

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

або декількох деталей) вхідна до складу механізму.

Група - кінематичний ланцюг, що складається з рухливих ланок, зв'язаних між собою кінематичними парами (відносинами), і задовольняючій деякій заданій умовам.

Вузол - кілька деталей зв'язаних між собою функціонально, конструктивно або яким-небудь іншим образом. З погляду системи вузли, групи, прості або типові механізми розглядаються як підсистеми. Найнижчим рівнем розбивки системи при конструюванні є рівень деталей ; при проектуванні - рівень ланок. Елементи із системи можна виділити тільки після визначення взаємозв'язків між ними, які описуються відносинами. Для механічних систем інтерес представляють відносини визначальну структуру системи і її функцій, тобто розташування й зв'язки.

Розташування - такі відносини між елементами, які описують їх геометричні відносні положення.

Зв'язки - відносини між елементами, призначені для передачі матеріалу, енергії або інформації між елементами. Зв'язки можуть здійснюватися за допомогою різних фізичних засобів: механічних з'єднань, рідин, електромагнітних або інших полів, пружних елементів. Механічні з'єднання можуть бути рухливими(кінематичні пари) і нерухомими. Нерухомі з'єднання діляться на рознімні (гвинтові, штифтові) і нероз'ємні (зварені, клейові).

Проектування складних об'єктів базується на застосуванні ідей і принципів, викладених у ряді теорій і підходів. Найбільш загальним підходом є *системний підхід*, ідеями якого пронизані різні методики проектування складних систем.

Основні поняття системотехніки

У теорії систем і системотехніці введений ряд термінів, серед них до базових потрібно віднести наступні поняття.

Система — множина елементів, що знаходяться у відносинах і зв'язках між собою.

Елемент — така частина системи, представлення про яку недоцільно піддавати при проектуванні подальшому} членуванню.

Складна система - система, характеризуєма великим числом елементів і. що найбільше важливо, великим числом взаємозв'язків елементів. Складність системи визначається також видом взаємозв'язків елементів, властивостями цілеспрямованості, цілісності, ієрархічності, багато - аспектності. Очевидно, що сучасні автоматизовані інформаційні системи і, зокрема, системи автоматизованого проектування, є складними в силу наявності в них перерахованих властивостей і ознак.

Підсистема — частина системи (підмножина елементів і їхніх взаємозв'язків), що має властивості системи.

Надсистема — система, стосовно якої розглянута система є підсистемою.

Структура — відображення сукупності елементів системи і їхніх взаємозв'язків: поняття структури відрізняється від поняття самої системи також тим, що при описі структури беруть до уваги лише типи елементів і зв'язків без конкретизації значень їхніх параметрів.

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

Параметр — величина, що виражає властивість або системи, або її частини, або середовища, що впливає на систему. Звичайно в моделях систем як параметри розглядають величини, що не змінюються в процесі дослідження системи. Параметри підрозділяють на зовнішні, внутрішні і вихідні, що виражають властивості елементів системи, самої системи, зовнішнього середовища відповідно. Вектори внутрішніх, вихідних і зовнішніх параметрів далі позначаються відповідно

$$\mathbf{X} = (x_1, x_2, \dots, x_n), \mathbf{Y} = (y_1, y_2, \dots, y_m), \mathbf{Q} = (q_1, q_2, \dots, q_k).$$

Фазова змінна — величина, що характеризує енергетичне або інформаційне наповнення елемента або підсистеми.

Стан — сукупність значень фазових перемінних, зафіксованих в одній часовій точці процесу функціонування,

Поводження (динаміка) системи — зміна стану системи в процесі функціонування.

Система без післядії — її поведінка при $t > 0$ визначається завданням стану в момент $t(j)$ і вектором зовнішніх впливів $Q(t)$. У системах з післядією, крім того, потрібно знати передісторію поведінки, тобто стану системи в моменти, що передують t_0 .

Вектор перемінних V , що характеризують стан (вектор перемінні стани). — ненадлишкова множина фазових перемінних, завдання значень яких у деякий момент часу цілком визначає поведінку системи надалі (в автономних системах без післядії).

Простір станів — множина можливих значень вектора перемінні стани.

Фазова траєкторія — представлення процесу (залежності $V\{Y\}$) у виді послідовності точок у просторі станів.

До характеристик складних систем, як сказано вище, часто відносять наступні поняття.

Цілеспрямованість — властивість штучної системи, що виражає призначення системи. Це властивість необхідна для оцінки ефективності варіантів системи.

Цілісність — властивість системи, що характеризує взаємозв'язок елементів і наявність залежності вихідних параметрів від параметрів елементів, при цьому більшість вихідних параметрів не є простим повторенням або сумою параметрів елементів.

Ієрархічність - властивість складної системи, що виражає можливість і доцільність її ієрархічного опису, тобто представлення у виді декількох рівнів, між компонентами яких мають відносини ціле-частина.

Складовими частинами системотехніки є наступні основні розділи:

- ієрархічна структура систем, організація їхнього проектування;
- аналіз і моделювання систем;
- синтез і оптимізація систем.

Моделювання має дві чітко помітні задачі:

- 1 — створення моделей складних систем (в англійському написанні — modeling);
- 2 — аналіз властивостей систем на основі дослідження їхніх моделей (simulation).

Синтез також підрозділяють на дві задачі:

- 1 — синтез структури проєктованих систем (структурний синтез);
- 2 — вибір чисельних значень параметрів елементів систем (параметричний синтез).

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

Ці задачі відносяться до області прийняття проектних рішень.

Моделювання й оптимізацію бажано виконувати з урахуванням статистичної природи систем. Детермінованість - лише окремий випадок. При проектуванні характерна недостача достовірних вихідних даних, невизначеність умов прийняття рішень. Облік статистичного характеру даних при моделюванні значною мірою заснований на методі статистичних іспитів (методі Монте-Карло), а прийняття рішень — на використанні нечітких множин, експертних систем, еволюційних обчислень.

Приклад 1. Комп'ютер є складною системою в силу наявності в нього великого числа елементів, різноманітних зв'язків між елементами і підсистемами, властивостей цілеспрямованості, цілісності, ієрархічності. До підсистем комп'ютера відносяться процесор (процесори), оперативна пам'ять, кеш-пам'ять, шини, пристрої введення-висновку.

Як надсистема може виступати обчислювальна мережа, автоматизована і (або) організаційна система, до яких належить комп'ютер. Внутрішні параметри — часи виконання арифметичних операцій, читання ('запису) у нагромаджувачах, пропускна здатність шин і ін. Вихідні параметри — продуктивність комп'ютера, ємність оперативної і зовнішньої пам'яті, собівартість, час наробітку на відмовлення й ін. Зовнішні параметри — напруга харчування мережі і його стабільність, температура навколишнього середовища й ін.

Приклад 2. Для двигуна внутрішнього згоряння підсистемами є колінчатий вал, механізм газорозподілу, поршнева група, система змащення й охолодження. Внутрішні параметри — число циліндрів, обсяг камери згоряння й ін. Вихідні параметри — потужність двигуна, КПД, витрата палива й ін. Зовнішні параметри — характеристики палива, температура повітря, навантаження на вихідному валові.

Приклад 3. Підсистеми електронного підсилювача — підсилювальні каскади: внутрішні параметри — опору резисторів, ємності конденсаторів, параметри транзисторів: вихідні параметри — коефіцієнт підсилення на середніх частотах, смуга пропускання, вхідний опір: зовнішні параметри — температура навколишнього середовища, напруги джерел харчування, опір навантаження.

Принципи системного підходу.

Основні ідеї і принципи проектування складних систем виражені в системному підході. Для фахівця в області системотехніки вони є очевидними і природними, однак їхнє дотримання і реалізація найчастіше пов'язані зі значними труднощами, що обумовлюються особливостями проектування. Інженери використовують системний підхід без звертання за допомогою до системного аналізу. Однак інтуїтивний підхід без застосування правил системного аналізу може виявитися недостатнім для рішення усе більш складних задач інженерної діяльності.

Основний загальний принцип системного підходу полягає в розгляді частин явища або складної системи з урахуванням їх взаємодії. Системний підхід містить у собі виявлення структури системи, типізацію зв'язків, визначення атрибутів, аналіз впливу зовнішнього середовища.

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

Системний підхід є базою для узагальнюючої дисципліни "Теорія систем" (інша використовувана назва - "Системний аналіз"). Теорія систем — дисципліна, у якій конкретизуються положення системного підходу; вона присвячена дослідженню і проектуванню складних економічних, соціальних, технічних систем, найчастіше слабо-структурованих. Характерними прикладами таких систем є виробничі системи. При проектуванні систем мети досягаються в багатокрокових процесах прийняття рішень. Методи прийняття рішень часто виділяють у самостійну дисципліну, названу "Теорія прийняття рішень".

Технічну дисципліну, в якій досліджуються складні технічні системи, їхнє проектування, і аналогічну теорії систем, часто називають системотехніка!

Предметом системотехніки є,

- по-перше, організація процесу створення, використання і розвитку технічних систем,
- по-друге, методи і принципи їхнього проектування і дослідження.

У системотехніці важливо вміти сформулювати мету системи й організувати її розгляд з позицій поставлених цілей. Тоді можна відкинути зайві і малозначимі частини при проектуванні і моделюванні, перейти до постановки оптимізаційних задач.

Системи автоматизованого проектування і керування відносяться до числа найбільш складних сучасних штучних систем. Їхнє проектування і супровід неможливі без системного підходу. Тому, ідеї і положення системотехніки входять складовою частиною в дисципліни, присвячені вивченню сучасних автоматизованих систем і технологій їхнього застосування. Інтерпретація і конкретизація системного підходу мають місце в ряді відомих підходів з іншими назвами, що також можна розглядати як компоненти системотехніки. Такими є структурні, блочно-ієрархічні, об'єктно-орієнтовані підходи.

При структурному підході, як різновиду системного, потрібно синтезувати варіанти системи з компонентів (блоків) і оцінювати варіанти при їхньому частковому переборі з попереднім прогнозуванням характеристик компонентів.

Блочно-ієрархічний підхід до проектування використовує ідеї декомпозиції складних описів об'єктів і відповідно засобів їхнього створення на ієрархічні рівні й аспекти, вводить поняття стилю проектування (висхідне і спадне), устанавлює зв'язок між параметрами сусідніх ієрархічних рівнів.

Ряд важливих структурних принципів, використовуваних при розробці інформаційних систем і насамперед їхнього програмного забезпечення (ПО), виражений в *об'єктно-орієнтованому підході до проектування* (ООП). Такий підхід має наступні переваги в рішенні проблем керування складністю й інтеграції ПО: 1) вносить у моделі додатків велику структурну визначеність, розподіляючи представлені в додатку дані і процедури між класами об'єктів; 2) скорочує обсяг специфікацій, завдяки введенню в описи ієрархії об'єктів і відносин спадкування між властивостями об'єктів різних рівнів ієрархії; 3) зменшує імовірність перекручування даних унаслідок помилкових дій за рахунок обмеження доступу до визначених категорій даних в об'єктах. Опис у кожному класі об'єктів припустимих звертань до

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

них і прийнятих форматів повідомлень полегшує узгодження й інтеграцію ПО.

Для всіх підходів до проектування складних систем характерні також наступні особливості:

Структуризація процесу проектування, що виражається декомпозицією проектних задач і документації, виділенням стадій, етапів, проектних процедур. Ця структуризація є сутністю блочно-ієрархічного підходу до проектування.

Ітераційний характер проектування.

Типізація й уніфікація проектних рішень і засобів проектування.

Особливості проектування СОС

Системний підхід до проектування

Дослідження об'єктів проектування за допомогою їхніх математичних моделей складає суть системного підходу. Виділяють наступні принципи системного підходу:

Ієрархічність - Кожна система або елемент може розглядатися як окрема система.

Структурність - Складається в можливості опису системи через опис комутаційних зв'язків між її елементами.

Взаємозалежність - Полягає в прояві властивостей системи тільки при взаємодії з зовнішнім середовищем.

Множинність опису - Полягає в описі системи на основі множини взаємодіючих математичних моделей.

Цілісність - Властивості всієї системи визначаються на основі аналізу властивостей її частин.

Суть системного підходу – це проектування частини з урахуванням цілого.

Ієрархічні багаторівневі системи, як об'єкт проектування

Декомпозиція технічних СОС.

Системне проектування нових технічних СОС

Виявлення загальних закономірностей проектно-конструкторського процесу, виділення загальних етапів і процедур, розробка різних методів рішення задач на цих етапах - справа, необхідна як для стимулювання творчої діяльності досвідчених проектувальників або навчання молодих фахівців, так і для підвищення ефективності самого процесу автоматизації.

Схема системного проектування СОС, в заг. випадку, являє собою систему дій інженера

38 Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

ч.І

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

від задуму технічного засобу до його втілення в комплект проектно-конструкторської документації.

Схема реалізовує наступні основні положення системного підходу до процесу проектування:

- розробка проекту йде від загального до конкретного, а не навпаки, тому проектувальнику слід братися за вирішення конкретних задач, тільки проробивши загальні;
- при розробці конкретних задач необхідно враховувати технічні рішення (ТР), прийняті на більш ранніх етапах проектування;
- нові ТР з'являються внаслідок творчого процесу, що носить ітераційний характер послідовних наближень до цілі;
- раціональне ТР досягається шляхом розробки максимального числа варіантів і їх поглибленим аналізом;
- при прийнятті рішення вимоги оптимального функціонування СОС переважають над іншими, наприклад, економічними;
- граничні конструктивно-технологічні параметри СОС диктуються лише фізико-технічними, а не економічними чинниками, тому при проектуванні спочатку проробляють інженерні розрахунки;
- проектування СОС обов'язково виконують з урахуванням можливості і трудомісткості їх виготовлення;
- економічна оцінка конструкції завжди є важливим стимулом отримання раціональних рішень, але може бути зроблена не раніше, ніж з'являться варіанти, що відповідають вимогам функціонування виробу і технічно здійсненні;
- при проектуванні необхідно максимально використати відомі технічні рішення, тобто існуючі прототипи СОС;
- при прийнятті рішень проектувальник повинен враховувати весь комплекс критеріїв укладених в таких показниках якості СОС, як функціонування, надійність, технологічність, стандартизація і уніфікація, а також ергономічні, естетичні і економічні показники;
- патентно-правові показники - необхідні критерії оцінки нових конкурентно - здатних технічних рішень;
- при проектуванні нових СОС потрібно передбачати також їх безболісну ліквідацію після закінчення терміну служби.

Системне проектування нових СОС складається з 4-х етапів:

- 1) постановки задачі створення нового технічного засобу,
- 2) пошукового проектування,
- 3) концептуального проектування і
- 4) інженерного конструювання.

На етапі постановки задачі створення нового технічного засобу на основі з'ясування і поглибленого аналізу проблеми виникнення актуальної потреби формується системна модель нового виробу, що описує його зв'язки і відносини із зовнішнім оточенням (рис. 1).



Рис. 1. Схема сист. Проект-ня СОС: 1- постановка задачі.

Розгляд цієї моделі дозволяє здійснити постановку загальної задачі створення нового СОС.

На етапі постановки задачі формулюється його службове призначення, визначаються обмеження і граничні умови на реалізацію робочої функції, критерії оцінки і т.д.

При аналізі задачі на новизну і технічну можливість реалізації визначаються шляхи подальшого ходу її рішення: використання існуючого технічного рішення, конструювання

нового технічного засобу або повторний розгляд проблеми з постановкою реальних на сьогоднішній день задач. Даний етап повинен відповісти на питання: чи потрібний новий СОС і які задачі він повинен вирішувати. При позитивному розв'язанні цих питань оформляється завдання, в якому остаточно формулюється постановка загальної задачі створення нового виробу, яке є основою для виконання етапів проектно-конструкторського процесу.



Рис. 2. Схема сист. Проект-ня СОС. 2- пошукове проектування.

Етап пошукового проектування повинен відповісти на питання - яким повинен бути майбутній технічний СОС (рис. 2). Для цього уточнюється його службове призначення, визначаються границі системи і її зв'язки із зовнішнім оточенням. При аналізі загальної задачі чітко формулюється робоча функція СОС і визначаються компоненти задачі - параметри, чинники рішення, цілі і критерії оцінки, час що відводиться на виконання проекту. Визначається (вибирається або знаходиться) принцип дії майбутнього технічного об'єкта.

Якщо на сьогоднішній день задача створення нового СОС виявиться технічно нездійсненна, то необхідно повернутися до постановки задачі його створення, уточнивши або змінивши його службове призначення. Коли принцип дії ясний і робоча схема об'єкта, що створюється відома, то потрібно визначити граничні режими функціонування об'єкта проектування.

Результатом даного етапу є оформлене технічне завдання на проектування нового СОС, яке повинно містити однозначний опис його службового призначення, показників якості і критерії оцінки проекту.

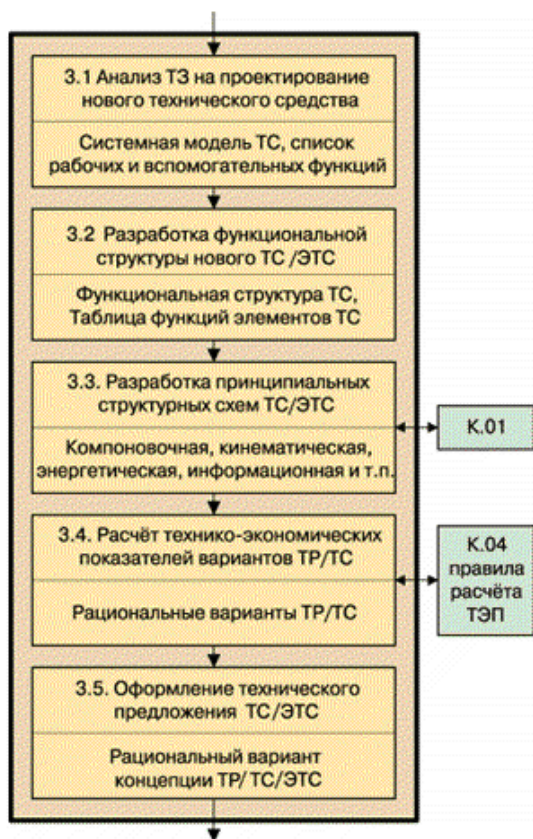


Рис. 3. Схема сист. Проект-ня СОС. 3- концептуальное проектирования

Етап концептуального проектування вирішує питання про технічну реалізацію задуму майбутньої конструкції (рис. 3). Розробка і аналіз різних варіантів принципових рішень (функціональної, компоновочної, кінематичної і інших схем) дає концепцію конструкції.

На цьому етапі проводиться економічна оцінка відібраних варіантів. Результатом етапу концептуального проектування повинна бути оформлена технічна пропозиція, яку повинна визначити концепцію конструкції майбутнього технічного засобу і техніко-економічну доцільність його створення.

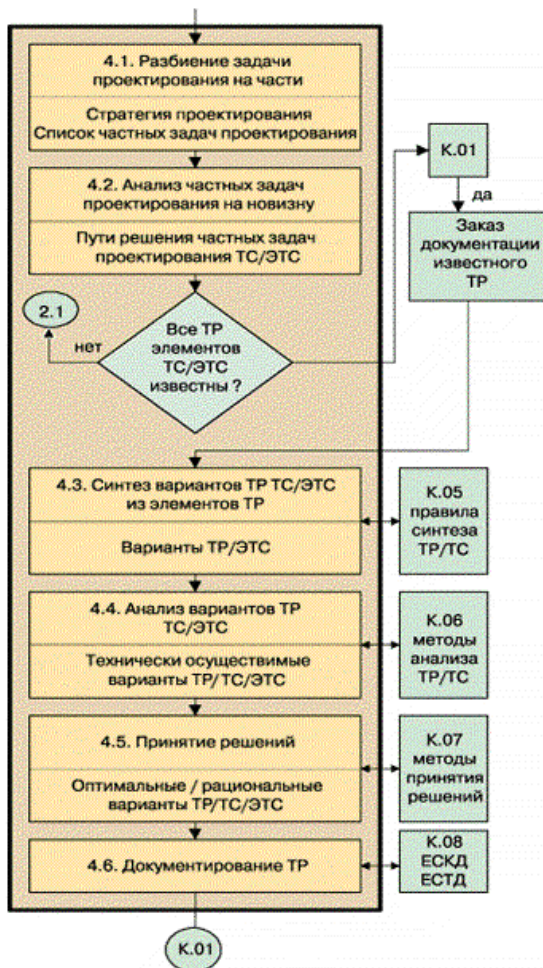


Рис. 4. Схема сист. Проект-ня СОС: 4- інженерне проектування

На етапі інженерного проектування (рис. 4) розробляються варіанти найважливіших елементів технічного засобу (ЕТЗ), які аналізуються і уточнюються (ескізне конструювання).

Потім виконується технічне і робоче проектування, які дають повне і остаточне уявлення про пристрій і функціонування майбутнього виробу, передбачають деталізацію конструкції СОС шляхом розробки креслень на кожний елемент, що виготовляється.

Об'єм комплексу конструкторської документації повинен відповісти на питання - яким повинен бути майбутній СОС і як насправді він працює, як його ремонтувати, як транспортувати, тощо .

На схемах показані і елементи необхідної інформаційної підтримки проектно-конструкторського процесу. Вони являють собою:

- каталоги відомих технічних рішень технічних засобів і їх елементів (К.01),
- довідники по фізичних ефектах, методах і способах перетворення речовини, енергії і інформації (К.02 і К.03),
- збірники випробуваних правил синтезу технічних рішень для технічних засобів різних видів (К.05),
- збірники випробуваних методів аналізу варіантів технічних рішень (К.06) і методів прийняття рішень (К.07) на різних стадіях проектування,
- опис правил розрахунку техніко-економічних показників (ТЭП), що рекомендуються нових технічних засобів і їх елементів (К.04).

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

Оформлення документації повинно виконуватися на вимоги ЕСКД і ЕСТД.

Етапи постановки загальної задачі створення нових технічних об'єктів, їх пошукового і концептуального проектування залишаються в основному етапами творчими. Методи їх автоматизації, що пропонуються деякими авторами зводяться до рекомендацій по створенню розвинених інформаційних і експертних систем. Найбільші успіхи досягнуті головним чином при автоматизації процесу інженерного конструювання. Але створення САПР і систем автоматизованої підтримки інженерних рішень повинно спиратися на докладний опис об'єкта і процесу його проектування.

3. САПР в автоматизованому проектуванні СОС

Склад, структура та класифікація САПР

Кожна система автоматизованого проектування складається зі складових, які називають «забезпеченням». Всього виділяють 7 видів забезпечень САПР.

Види забезпечення САПР

1) Математичне забезпечення - сукупність математичних методів і моделей, необхідних для виконання процесу автоматизованого проектування.

2) Програмне забезпечення - сукупність машинних програм, представлених у заданій формі, разом із програмною документацією, необхідних для здійснення процесу проектування, що включає системне і прикладне ПО.

3) Технічне забезпечення - сукупність взаємозалежних і взаємодіючих технічних засобів для введення, збереження, переробки, передачі програм і даних, організації спілкування оператора з ЕОМ, виготовлення проектної документації.

4) Лінгвістичне забезпечення – сукупність мов проектування, включаючи терміни, визначення, правила формалізації природної мови, методи стиску і розгортання текстів.

5) Інформаційне забезпечення - сукупність даних і знань, які представлені у певній формі та необхідні для виконання автоматизованого проектування, у тому числі для опису стандартних проектних процедур, типових проектних рішень, типових елементів, що комплектують виробів, матеріалів і ін. Включає СУБД (Система Керування Базами Даних), саму базу даних і базу знань.

6) Методичне забезпечення - документи, у яких визначені склад, правила добору й експлуатації засобів автоматизації проектування

7) Організаційне забезпечення - сукупність документів, що визначають склад проектної організації і її підрозділів, їхньої функції, зв'язку між ними і комплексом засобів автоматизації, а також форму представлення результатів проектування і порядок розгляду проектних документів

Класифікація САПР

Класифікацію САПР здійснюють по ряду ознак, наприклад, по додатку, цільовому призначенню, масштабам (комплексності розв'язуваних задач), характерові базової підсистеми — ядра САПР..

САПР класифікують по:

- різновиду і складності об'єктів проектування:

а) САПР низько складних об'єктів (кількість складових частин до 100);

б) САПР середньо складних об'єктів (100-10 000);

в) САПР високо складних об'єктів (вище 10 000).

- рівневі автоматизації:

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

- а) низкоавтоматизированные (до 25% проектних процедур автоматизоване);
 - б) среднеавтоматизированные (25%-50%);
 - в) високоавтоматизовані (50%-75%).
- рівневі комплексності:
- а) одноетапні (один етап проектування);
 - б) многоэтапные (кілька етапів);
 - в) комплексні (весь процес створення виробу).
- характерові і числу документів, що випускаються проектом,:
- а) САПР низкою продуктивності (100-10 000 проектних документів у перерахуванні на формат А4 за рік);
 - б) САПР середньої продуктивності (10 000- 100 000);
 - в) САПР високої продуктивності (100 000 і вище).
- За ДСТУ інформація повинна зберігатися як у твердому виді (на папері) так і на магнітному носії (магнітна стрічка); термін збереження – 50 років.
- числу рівнів технічного забезпечення:
- а) одноуровневый (на основі ЕОМ середнього і високого класу зі штатним периферійним устаткуванням);
 - б) дворівневий (на основі ЕОМ середнього і високого класу, як інтелектуальні термінали – персональні ЕОМ);
 - в) комплексний (на основі ЕОМ середнього і високого класу, об'єднаних у мережу і кожна з ЕОМ має мережа інтелектуальних терміналів на основі персональних ЕОМ).

По додатках найбільш представницькими і широко використовуваними є наступні групи САПР.

САПР для застосування в галузях загального машинобудування. Їх часто називають машинобудівними САПР або МСAD (Mechanical CAD) системами.

САПР для радіоелектроніки. Їхньої назви — ЕСAD (Electronic CAD) або EDA (Electronic Design Automation) системи.

САПР в області архітектури і будівництва.

Крім того, відомо велике число більш спеціалізованих САПР, або виділюваних у зазначених групах, або представляючих самостійну галузь у класифікації. Прикладами таких систем є САПР великих інтегральних схем (БІС): САПР літальних апаратів: САПР електричних машин і т.п.

По функціональному призначенню розрізняють САПР або підсистеми САПР. різні аспекти, що забезпечують, (страсти) проектування. Так. у складі МСAD з'являються САЕ/CAD/CAM системи :

САПР функціонального проектування, інакше САПР-Ф або САЕ (Computer Aided Engineering) системи.

конструкторські САПР загального машинобудування САПР-К, часто називані просто САD системами;

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

технологічні САПР загального машинобудування — САПР-Т. інакше називані автоматизованими системами технологічної підготовки виробництва АСТПП або системами САМ (Computer Aided Manufacturing).

По масштабах розрізняють окремі програмно-методичні комплекси (ПМК) САПР, наприклад, комплекс аналізу міцності механічних виробів відповідно до методу кінцевих елементів (МКЭ) або комплекс аналізу електронних схем; системи ПМК: системи з унікальними архітектурами не тільки програмного (software), але і технічного (hardware) забезпечень.

По характеру базової підсистеми розрізняють наступні різновиди САПР.

1. САПР на базі підсистеми машинної графіки і геометричного моделювання. Вони орієнтовані на додатки, де основною процедурою проектування є конструювання, тобто визначення просторових форм і взаємного розташування об'єктів. Тому до цієї групи систем відноситься більшість графічних ядер САПР в області машинобудування.

В даний час з'явилися уніфіковані графічні ядра, застосовувані більш ніж в одній САПР, це ядра Parasolid фірми EDS Unigraphics і ACIS фірми Intergraph.

САПР на базі СУБД. Вони орієнтовані на додатки, у яких при порівняно нескладних математичних розрахунках переробляється великий обсяг даних. Такі САПР переважно зустрічаються в техніко-економічних додатках, наприклад, при проектуванні бізнес-планів, але мають місце також при проектуванні об'єктів, подібних до щитів керування в системах автоматики.

САПР на базі конкретного прикладного пакета. Фактично це автономно використовувані програмно-методичні комплекси, наприклад, імітаційного моделювання виробничих процесів, розрахунку міцності по методу кінцевих елементів, синтезу й аналізу систем автоматичного керування і т.п. Часто такі САПР відносяться до систем САЕ. Прикладами можуть служити програми логічного проектування на базі мови VHDL, математичні пакети типу MathCAD.

Комплексні (інтегровані) САПР, що складаються із сукупності підсистем попередніх видів. Характерними прикладами комплексних САПР є САЕ/CAD/CAM-системи в машинобудуванні або САПР БІС. Так, САПР БІС' містить у собі ~34816 і підсистеми проектування компонентів, принципів, логічних і функціональних схем, топології кристалів, тестів для перевірки придатності виробів. Для керування настільки складними системами застосовують спеціалізовані системні середовища.

Стадії створення проекту в САПР

При створенні проекту в САПР розрізняють зовнішнє і внутрішнє проектування.

До зовнішнього проектування відносять:

- Стадію перед-проектних досліджень
- Розробку технічного завдання на проект

Внутрішнє проектування включає:

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

- розробку технічних пропозицій
- ескізний проект
- технічний проект
- робочий проект
- виготовлення, налагодження, тестування і введення системи в дію

Дані стадії можуть бути застосовують для проектування більшості технічних об'єктів і систем.

Типова функціональна структура САПР

Складовими структурними частинами САПР є підсистеми, що володіють усіма властивостями системи і які створюються як самостійні системи.

Підсистеми САПР

Підсистемою САПР називають виділену по деяких ознаках частину *САПР*, що дозволяє одержувати закінчені проектні рішення.

У САПР виділяють *проектуючі* і *обслуговуючі* підсистеми. Підсистеми, що проектують, виконують проектні процедури й операції. Обслуговуючі підсистеми призначені для підтримки працездатності об'єктно-орієнтованих підсистем і виконуються процедури й операції, безпосередньо зв'язані з конкретним типом об'єктів проектування. В об'єктно-незалежних - уніфіковані процедури й операції.

У залежності від відношення до об'єкта проектування, підсистеми поділяють на:

- - об'єктно-орієнтовані,
- - об'єктно-незалежні.

До проектуючих підсистем відносять:

- Підсистему функціонально-логічного проектування - на виході цієї системи ми одержуємо функціональну схему, за нею логічну схему, на виході принципово - електричну схему.
- Підсистему конструкторського проектування - на виході одержуємо конструкцію пристрою і конструкторську документацію, що включає схему розташування елементів на поверхні модуля і топологію друкованих з'єднань між елементами.
- Підсистему технологічного проектування - на виході одержуємо маршрутну карту виробничого процесу і програми для керування верстатів з числовим програмним керуванням (для керування технологічним устаткуванням).

До обслуговуючих підсистем відносять:

- Систему інформаційного пошуку
- Систему документування
- Систему графічного відображення об'єктів проектування

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

До складу як проектуючих, так і обслуговуючих систем сучасних САПР можуть входити:

Експертні системи. Це системи, в основі яких лежить база знань, представлена або у виді системи продукції, або у виді фреймів (*FRAME*). Експертна система дозволяє формалізувати знання експерта у визначеній предметній області з метою прийняття раціональних проектних рішень.

Системи прийняття рішень. Це системи, що дозволяють робити вибір ефективних проектних рішень в умовах визначеності і невизначеності вихідної інформації на основі формальних методів і процедур. Для оцінки проектних рішень можуть також застосовуватися нейросетевые технології.

Експертні системи в САПР

Експертна система САПР використовується для рішення задач автоматизованого проектування і є, у свою чергу, обчислювальною системою. Вона повинна задовольняти наступним вимогам:

- Прийняті, за допомогою системи, рішення повинні відповідати рівневі експерта-професіонала.
- Способи прийняття рішень (мета-судження) у будь-який момент часу повинні відтворюється у формі, зрозумілій як експертові, так і користувачеві.
- Система повинна адаптуватися до користувача за рахунок можливості змінювати як формулювання запитів і задач, так і послідовність їхнього виникнення.
- Система повинна мати можливість використовувати, здобувати і зберігати загальні й локальні схеми міркування, побудовані на нецілком достовірних даних і символічних перетвореннях.

У процесі життєвого циклу система повинна мати властивість ревізії даних і схем міркувань.

До переліку основних компонентів ЕС САПР входять:

- Лінгвістичний процесор
- Підсистема логічного висновку.
- База знань.
- Підсистема ревізії знань.
- Робоча пам'ять.

На рис. показано структурну схему узагальненої експертної системи.

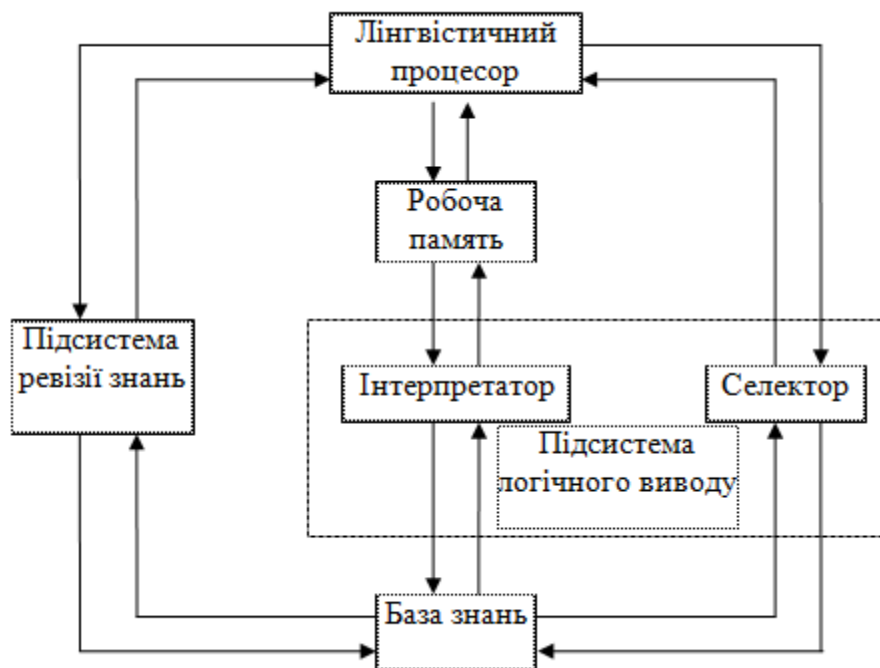


Рис.Х. Структура узагальненої експертної системи.

Лінгвістичний процесор здійснює зв'язок інших компонентів з користувачем або експертом алгоритмічною мовою.

Підсистема логічного висновку забезпечує побудову тієї або схеми міркування.

База знань призначена для збереження й обробки знань, представлених логічними, продукційними або семантичними моделями.

Підсистема ревізії знань дозволяє користувачеві або експертові втручатися в процес підготовки ухвалення рішення за рахунок пояснення (відображення) проміжних дій у системі.

Робоча пам'ять забезпечує збереження проміжних даних і їхній обмін між компонентами системи.

У деяких роботах по штучному інтелекту можна зустріти трохи інше представлення узагальненої експертної системи, причому, принциповою відмінністю може з'явитися наявність у структурі підсистеми придбання й інтерпретації знань. Однак у таких системах, як EURISKO, роль такої підсистеми виконує підсистема логічного висновку разом з підсистемою ревізії знань, а в системі MYSIN її неможливо виділити як окремий програмний засіб. У системах, побудованих за технологією "prototyping" – ИНТЕР-ЭКСПЕРТ (GURU), ЕКСПЕРТИЗА, тобто на основі оболонок, також важко виділити такий програмний модуль, що забезпечував би придбання знань.

Інтеграція САПР з іншими автоматизованими системами

Проектування нових виробів неможливо без застосування передових технологій, що використовують сучасні системи автоматизованого проектування. Тому, більшість компаній,

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

які розповсюджують та впроваджують САПР, пропонують єдиний програмно-апаратний комплекс, що реалізує наскрізну автоматизацію проектування і підготовки виробництва. Однак, побудова такого комплексу САПР – задача не проста: 1) потрібно розробити маршрути проектування; 2) погодити взаємодію різних модулів; 3) визначити необхідну кількість робочих місць для кожного етапу проектування; 4) адаптувати систему до конкретних вимог ОП та до проектного середовища; 5) налагодити таку систему проектування на тестових та пілотних проектах; 6) проробити стикування форматів технологічних файлів з відповідним виробничим і тестовим устаткуванням; 7) провести відповідне навчання персоналу, тощо. З іншого боку, компанії-виробники САПР також розширюють можливості своїх систем для забезпечення такого комплексного проектування. Тому, останнім часом на ринку САПР з'являються такі комплексні рішення, які позначаються англійською аббревіатурою CAD/CAM/CAE/CAPP/PDM.

Скорочення CAD/CAM/CAE/CAPP/PDM означає:

- Computer Aided Design - Система Автоматизованого Проектування (САПР);
- Computer Aided Manufacturing – Автоматизована система технологічної підготовки виробництва (АСТПВ);
- Computer Aided Engineering – Система автоматизованої підтримки інженерних рішень (САПР);
- Computer Aided Process Planning – Автоматизована система виробничого планування (АСВП);
- Product Data Management – Система управління проектами і технічним документообігом.
- Прикладом такої системи є російський комплекс програм T-FLEX.
- ...<дати характеристику комплексу>...

Для проведення задачі аналізу в даний час одержали широке поширення СППР (Системи Підтримки Прийняття Рішень), СПР і експертні системи, що будуються на принципах штучного інтелекту, в основі лежить продукційний підхід, семантичні мережі і фрейми, і мови штучного інтелекту (Prolog і Lisp). Фрейми – сукупності взаємозалежних даних, що дозволяють точно визначити характер об'єкта. СППР і експертні системи можуть містити в собі одночасно весь спектр вищевказаних моделей і засобів.

Існує безліч визначень СППР, що відбивають точки зору представників різних дисциплін і наукових шкіл. Так СППР може визначатися, як "заснована на використанні моделей сукупність процедур по обробці даних і суджень, що допомагають керівникові в прийнятті рішень".

У деяких випадках пропонується розглядати СППР у якості "інтерактивних автоматизованих систем, що допомагають особам, що приймають рішення, використовувати дані і моделі, щоб вирішувати неструктуризовані проблеми". СППР може виглядати як "комп'ютерна інформаційна система, використовувана для підтримки різних видів діяльності при прийнятті рішень у ситуаціях, де неможливо або небажано мати автоматичну систему, що

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

цілком виконує весь процес рішення". Більшість дослідників згодні, що СППР призначені для рішення слабоструктуризованих проблем.

У процесі прийняття рішень виникла необхідність у суб'єктивних, експертних моделях, що можуть бути вкрай корисні для ЛПР. Виникла також необхідність в обліку знань багатьох експертів, в аналізі прийнятих раніше рішень. У структурі СППР з'явився блок "база знань" (БЗ), і такі системи одержали назва "інтелектуальних".

Розвиток технічних програмних засобів, що дозволяють "індустріалізувати" технологію створення нових систем, привело до формування ще однієї точки зору на СППР, що одержала назву "адаптивного проектування". Прихильники цього підходу вважають, що термін СППР має право на існування тільки в тих випадках, коли "кінцева система" виникає в ході адаптивного процесу проектування і впровадження.

Місце Сапр серед інших автоматизованих систем.

Структура САПР. Як і будь-яка складна система. САПР складається з підсистем (рис. 1.1). Розрізняють підсистеми що проектують і обслуговують.

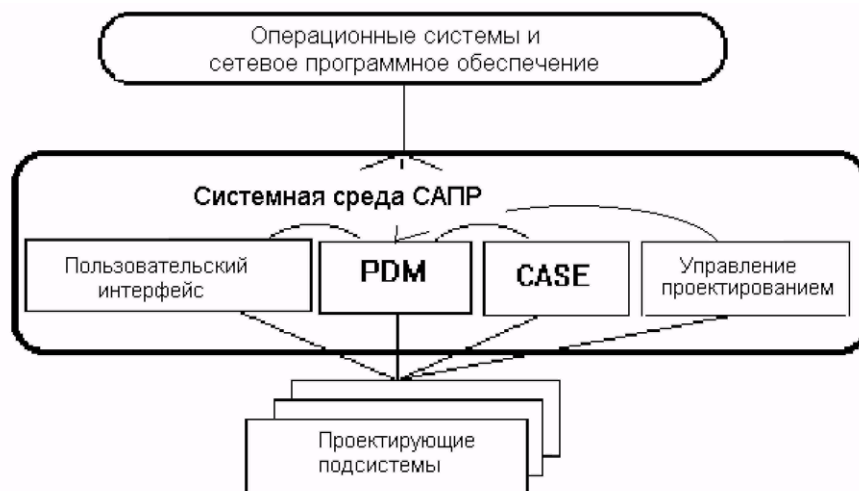


Рис 1.1. Структура програмного забезпечення САПР

Підсистеми, що проектують, безпосередньо виконують проектні процедури. Прикладами підсистем, що проектують, можуть служити підсистеми геометричного тривимірного моделювання механічних об'єктів, виготовлення конструкторської документації, схемотехнічного аналізу, трасування з'єднань у друкованих платах.

Обслуговуючі підсистеми забезпечують функціонування підсистем, що проектують, їхню сукупність часто називають системним середовищем (або оболонкою) САПР. Типовими обслуговуючими підсистемами є підсистеми керування проектними даними (PDM — Product Data Management), керування процесом проектування (DesPM — Design Process Management), користувацького інтерфейсу для зв'язку розроблювачів з ЕОМ. CASE (Computer Aided Software Engineering) для розробки і супроводу програмного забезпечення САПР, що навчають підсистеми для освоєння користувачами технологій, реалізованих у САПР. Структурування САПР по різних аспектах обумовлює поява розколбоків забезпечення ~43008. Прийнято виділяти сім видів забезпечення:

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

технічне (ТЕ), що включає різні апаратні засоби (ЕОМ. периферійні пристрої, мережне комутаційне устаткування, лінії зв'язку, вимірювальні засоби):

математичне (МО), що поєднує математичні методи, моделі й алгоритми для виконання проектування:

програмне (ПО), що представляється комп'ютерними програмами САПР:

— інформаційне (ІО). складається з баз даних БД), систем керування базами даних (СУБД), а також інших даних, використовуваних при проектуванні: відзначимо, що вся сукупність використовуваних при проектуванні даних називається інформаційним фондом САПР, а БД разом з СУБД зветься банку даних (БнД):

лінгвістичне (ЛО), що виражається мовами спілкування між проектувальниками й ЕОМ. мовами програмування і мовами обміну даними між технічними засобами САПР:

методичне (МетО). включає різні методики проектування, іноді до МетО відносять також математичне забезпечення:

організаційне (ОО). представляємо штатними розкладами, посадовими інструкціями й іншими документами, що регламентують роботу проектного підприємства.

Функції, характеристики і приклади САЕ/CAD/CAM-систем.

Функції CAD-систем у машинобудуванні підрозділяють на функції двовимірного (2D) і тривимірного (3D) проектування. До функцій 2D відносяться креслення, оформлення конструкторської документації: до функцій 3D — одержання тривимірних моделей, метричні розрахунки, реалістична візуалізація, взаємне перетворення 2D і 3D моделей.

Серед CAD-систем розрізняють "легкі" і "важкі" системи. Перші з них орієнтовані переважно на ІD графікові, порівняно дешеві і менш вимогливі у відношенні обчислювальних ресурсів. Другі орієнтовані на геометричне моделювання (3D), більш універсальні, дороги, оформлення креслярської документації в них звичайно здійснюється за допомогою попередньої розробки тривимірних геометричних моделей.

Основні функції Сам-систем: розробка технологічних процесів, синтез керуючих програм для технологічного устаткування з числовим програмним керуванням (ЧПУ). моделювання процесів обробки, у тому числі побудова траєкторій відносного руху інструмента і заготівлі в процесі обробки, генерація постпроцесорів для конкретних типів устаткування з ЧПУ (NC — Numerical Control), розрахунок норм часу обробки.

Найбільш відомі наступні САЕ/CAD/CAM-системи, призначені для машинобудування.

"Важкі" системи (у дужках зазначена фірма, що розробила або поширює продукт): Unigraphics (EDS Unigraphics): Solid Edge (Intergraph): **Pro/Engineer** (PTC — Parametric Technology Corp.). CATIA (Dassault Systemes). EUCLID (Matra Datavision). CADD5.5 (Coinpitervision. нині входить у PTC) і ін.

"Легкі" системи: AutoCAD (Autodesk): АДЕМ: bCAD : Caddy (Ziegler Informatics): Компас (Аскон. С.Петербург): Спрут (Spnit Technology. Набережні Челны): Кредо (ННВЦ АС'К. Москва).

Системи, що займають проміжне місце: Cimatixm. Micro station (Bentley). Euclid Prelude (Matra Datavision). T-FlexCAD (Топ Системи. Москва) і ін. З ростом можливостей персональних комп'ютерів, відмінності між "важкими" і "легкими" САПР поступово

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

стираються.

Функції CAE-систем досить різноманітні, тому що зв'язані з проектними процедурами аналізу, моделювання, оптимізації проектних рішень. До складу машинобудівних CAE-систем насамперед включають програми для наступних процедур:

моделювання полів фізичних величин, у тому числі аналіз міцності, що частіше за все виконується відповідно до MCE:

розрахунок станів і перехідних процесів на макрорівні:

імітаційне моделювання складних виробничих систем на основі моделей масового обслуговування і мереж Петрі.

Приклади систем моделювання полів фізичних величин відповідно до MCE: Nastran. Ansys. Cosmos. Nisa. Moldflow.

Приклади систем моделювання динамічних процесів на макрорівні: Adams і Dyna — у механічних системах. Spice — в електронних схемах. ПА9 — для багатоаспектного моделювання, тобто для моделювання систем, принципи дії яких засновані на взаємовпливі фізичних процесів різної природи.

Для зручності адаптації САПР до нестатків конкретних додатків, для її розвитку доцільно мати в складі САПР інструментальні засоби адаптації і розвитку. Ці засоби представлені тією або іншою CASE-технологією, включаючи мови розширення. У деяких САПР застосовують оригінальні інструментальні середовища.

Прикладами можуть служити об'єктно-орієнтоване інтерактивне середовище CAS.CADE у системі EUCLID, що містить бібліотеки компонентів. У САПР T-Flex CAD 3D передбачена розробка доповнень у середовищах Visual C++ і Visual Basic.

Важливе значення для забезпечення відкритості САПР, її інтегрованості з іншими автоматизованими системами (АС) мають інтерфейси, що представляються реалізованими в системі форматами міжпрограмних обмінів. Очевидно, що, у першу чергу, необхідно забезпечити зв'язку між CAE, CAD і CAM підсистемами.

Щодо форматів для міжпрограмних обмінів — використовуються IGES, DXF, Express (стандарт ISO 10303-11, совокупність стандартів STEP, SAT (формат ядра ACIS) та ін.

Найбільш перспективними вважаються діалекти мови Express, що порозумівається загальним характером стандартів STEP, їхньою спрямованістю на різні додатки, а також на використання в сучасних розподілених проектних і виробничих системах. Дійсно, такі формати, як IGES або DXF, описують тільки геометрію об'єктів, у той час як в обмінах між різними САПР і їхніми підсистемами фігурують дані про різні властивості й атрибути виробів.

Мова Express використовується в багатьох системах інтерфейсу між CAD/CAM-системами. Зокрема, у систему CAD++STEP включене середовище SDAI (Standard Data Access Interface), у якій можливе представлення даних про об'єкти з різних систем CAD і додатків (але описаних за правилами мови Express). CAD++ STEP забезпечує доступ до баз даних більшості відомих САПР із представленням даних, що витягається, у виді STEP-файлів. Інтерфейс програміста дозволяє відкривати і закривати файли проектів у базах даних, робити читання і запис сутностей. Як об'єкти можуть використовуватися точки, криві, поверхні, текст, приклади проектних рішень, розміри, зв'язки, типові зображення, комплекси даних і т.п.

54 Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

ч.І

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

Поняття про CALS-технології.

CALS-технологія — це технологія комплексної комп'ютеризації сфер промислового виробництва, ціль якої — уніфікація і стандартизація специфікацій промислової продукції на всіх етапах її життєвого циклу. Основні специфікації представлені проектною, технологічною, виробничою, маркетинговою, експлуатаційною документацією. У CALS-системах передбачені збереження, обробка і передача інформації в комп'ютерних середовищах, оперативний доступ до даних у потрібний час і в потрібному місці. Відповідні системи автоматизації назвали автоматизованими логістичними системами або CALS (Computer Aided Logistic Systems). Оскільки під логістикою звичайно розуміють дисципліну, присвячену питанням постачання і керування запасами, а функції CALS набагато ширше і зв'язані з всіма етапами життєвого циклу промислових виробів, застосовують і більш відповідну предметові розшифровку абревіатури CALS — Continuous Acquisition and LifeCycle Support.

Застосування CALS дозволяє істотно скоротити обсяги проектних робіт, тому що опису багатьох складених частин устаткування, машин і систем, що проектувалися раніше, зберігаються в базах даних мережних серверів, доступних будь-якому користувачеві технології CALS. Істотно полегшується рішення проблем ремонтпригодності, інтеграції продукції в різного роду системи і середовища, адаптації до мінливих умов експлуатації, спеціалізації проектних організацій і т.п. Очікується, що успіх на ринку складної технічної продукції буде немислимий поза технологією CALS.

Розвиток CALS-технології повинен привести до появи так званих віртуальних виробництв, при яких процес створення специфікацій з інформацією для програмно керованого технологічного устаткування, достатньої для виготовлення виробу, може бути розподілений у часі і просторі між багатьма організаційно автономними проектними студіями. Серед безсумнівних досягнень CALS-технології слід зазначити легкість поширення передових проектних рішень, можливість багаторазового відтворення частин проекту в нових розробках і ін.

Побудова відкритих розподілених автоматизованих систем для проектування і керування в промисловості складає основу сучасної CALS-технології. Головна проблема їхньої побудови — забезпечення однакового опису й інтерпретації даних, незалежно від місця

і часу їхнього одержання в обшій системі, що має масштаби аж до глобальних. Структура проектної, технологічної й експлуатаційної документації, мови її представлення повинні бути стандартизованими. Тоді стає реальною успішна робота над загальним проектом різних колективів, розділених у часі і просторі і використовуючих різні CAE/CAD/CAM-системи. Та сама конструкторська документація може бути використана багаторазово в різних проектах, а та сама технологічна документація адаптована до різних виробничих умов, що дозволяє істотно скоротити й здешевити загальний цикл проектування і виробництва. Крім того, спрощується експлуатація систем.

Отже, інформаційна інтеграція є невід'ємною властивістю CALS-систем. Тому в основу CALS-технології покладений ряд стандартів, що забезпечують таку інтеграцію.

Важливі проблеми, що вимагають рішення при створенні комплексних САПР — керування складністю проектів і інтеграція ПО. Ці проблеми включають питання декомпозиції проектів, розпаралелювання проектних робіт, цілісності даних, міжпрограмних інтерфейсів і

Комплексні автоматизовані системи.

Відомо, що часткова автоматизація найчастіше не дає очікуваного підвищення ефективності функціонування підприємств. Тому кращим є впровадження інтегрованих САПР, що автоматизує всі основні етапи проектування виробів. Подальше підвищення ефективності виробництва і підвищення конкурентоздатності продукції, що випускається, можливо за рахунок інтеграції систем проектування, керування і документообігу.

Така інтеграція лежить в основі створення комплексних систем автоматизації, у яких крім функцій проектування реалізуються засоби для автоматизації функцій керування проектуванням, документообігу, планування виробництва, обліку і т.п.

Очевидно, що подібна інтеграція є **невід'ємною** рисою CALS-систем. В основу CALS-технології покладений ряд стандартів і насамперед це стандарти STEP а також Parts Library. Mandate. SGML (Standard Generalized Markup Language). EDIFACT (Electronic Data Interchange For Administration. Commerce. Transport) і ін. Стандарт SGML установлює способи уніфікованого оформлення документів визначеного призначення — звітів, каталогів, бюлетенів і т.п., а стандарт EDIFACT — способи обміну подібними документами.

Одна з найбільш відомих реалізацій CALS-технологія розроблена фірмою Computer vision. Це технологія названа EPD (Electronic Product Definition) і орієнтована на підтримку процесів проектування й експлуатації виробів машинобудування.

У CALS-системах на всіх етапах життєвого циклу виробів використовується документація, отримана на етапі проектування. Тому природно, що склади підсистем у CALS і комплексних САПР значною мірою збігаються.

Технологію EPD реалізують:

CAD — система автоматизованого проектування:

CAM — автоматизована система технологічної підготовки виробництва (АСТПП):

CAE — система моделювання і розрахунків:

CAPE (Concurrent Art-to-Product Environment) — система підтримки рівнобіжного проектування (concurrent engineering):

PDM — система керування проектними даними, що представляє собою спеціалізовану ~34816 (DBMS — Data Base Management System):

3D Viewer -система тривимірної візуалізації:

CADD — система документування:

CASE — система розробки і супроводу програмного забезпечення:

методики обстеження й аналізу функціонування підприємств.

Основу EPD складають системи CAD і PDM. у якості яких використовуються CADDS5 і Ortegra відповідно. Значною мірою специфіку EPD визначає система Ortegra. У ній відображається ієрархічна структура іздєлій. включаюча всі складальні вузли і деталі. У Ortegra можна одержати інформацію про атрибути будь-якого елемента структури, а також відповіді на типові для баз дані питання типу "Укажіть деталі з матеріалу А" або "У яких блоках

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

використовуються деталі виготовлювача В?" і т.п.

Важливою для користувачів особливістю Optegra є робота разом із багатовіконною системою візуалізації 3D Viewer. Користувач може одночасно стежити за інформацією в декількох типових вікнах:

інформаційний браузер, у якому висвічуються дані, запитувані користувачем, наприклад, із поштової шухляди. Internet, корпоративних ресурсів, його персональної САПР:

вікно структури виробу, що представляється у виді дерева. Можна одержувати відповіді на запити підсвічуванням деталей Dj (листів дерева), що задовольняють умовам запиту:

3D візуалізатор. у цьому вікні висвічується тривимірне зображення виробу, відповіді на запити даються й у цьому вікні колірним виділенням деталей D;

вікно користувальницького процесу, у якому в потрібній послідовності у виді іконок відображається перелік задач, заданий користувачеві для рішення.

У системі Optegra зв'язку між об'єктами задаються по протоколах стандартів STEP, зовнішній інтерфейс здійснюється через базу даних SDAI.

Системи керування в складі комплексних автоматизованих систем. Системи керування в промисловості, як і будь-які складні системи, мають ієрархічну структуру. Якщо розглядати підприємство як систему верхнього рівня, то наступними рівнями по спадній лінії будуть рівні заводу, цеху, виробничого устаткування. -Автоматизація керування реалізується за допомогою автоматизованих систем керування (АСУ).

Серед АСУ розрізняють автоматизовані системи керування підприємством (АСУП) і автоматизовані системи керування технологічними процесами (АСУТП). АСУП охоплює рівні від підприємства до цеху, АСУТП — від цеху і нижче (на рівні цеху можуть бути засоби й АСУП. і АСУТП).

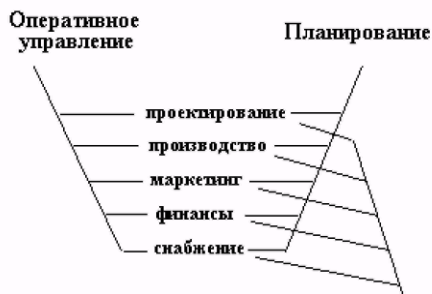


Рис 12 основні функції АСУП

В АСУП виділяють підсистеми, що виконують визначені функції (рис. 1.2), типовими серед них є:

- календарне планування виробництва,
- оперативне керування виробництвом:
- сіткове планування проектів:
- керування проектуванням виробів:
- облік і нормування трудозатрат:
- облік основних фондів:
- керування фінансами:

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

- керування запасами (складським господарством):
- керування постачанням (статистика закупівель, контракти на закупівлю):
- маркетинг (статистика й аналіз реалізації).
- контракти на реалізацію, прогноз, реклама).

Процедури, що виконують ці функції, часто називають бізнесами-функціями, а маршрути рішення задач керування, що складаються з бізнесів-функцій, називають **бізнесами-процесами**.

Примітка. Як сказано вище, у ~43008 аналогічні поняття називають проектними процедурами і маршрутами проектування.

Існують різновиду АСУП зі своїми англійськими назвами. Найбільш загальну систему з перерахованими вище функціями називають ERP (Enterprise Resource Planning). Системи, спрямовані на керування інформацією про матеріали, виробництво, контроль і т.п. виробів, називають MRP-2 (Manufacturing Resource Planning). У ERP, як і в ~43008, важлива роль приділяється системам керування даними PDM. Якщо PDM забезпечує керування конфігурацією проектів і відноситься в більшій мері до проектування, то MRP-2 керує даними, що відносяться до виробництва. Для таких систем іноді використовують також назва MES (Manufacturing Execution System).

Характерні риси сучасних АСУП.

Відкритість стосовно ведучих платформ (UNIX, Windows, OS/2) і різним СУБД і насамперед могутнім СУБД типу Oracle, Ingres, Informix, Sybase: підтримка технологій типу ODBC (Open Data Base Connection), OLE (Object Linking and Embedding), DDE (Dynamic Data Exchange): підтримка архітектур клієнт/сервер. Важлива характеристика — можливість роботи в середовищі розподілених обчислень.

Можливість наскрізного виконання всіх припустимих бізнесів-функцій або їхньої частини, що забезпечується модульною побудовою (кількість функцій може перевищувати 100).

Адаптованість до конкретних замовників і умов ринку.

Наявність інструментальних засобів, у тому числі мови розширення або 4GL (мови четвертого покоління). Так, у R3 використовується мова ABAP/L, у Elite Series — мова Informix-4GL.

Технічне забезпечення АСУП — комп'ютерна мережа, вузли якої розташовані як в адміністративних відділах підприємства, так і в цехах.

Очевидно, що для створення і розвитку віртуальних підприємств необхідне поширення CALS-технології не тільки на САПР, але і на АСУ, їхня інтеграція в комплексні системи інформаційної підтримки всіх етапів життєвого циклу промислової продукції.

Функціями АСУТП на рівнях цеху і ділянки є збір і обробка даних про стан устаткування і протікання виробничих процесів для прийняття рішень по завантаженню верстатів, по виконанню технологічних маршрутів. Програмне забезпечення АСУТП на цих рівнях представлено системою диспетчерського керування і збору даних, названої SCADA (Supervisory Control and Data Acquisition), а технічне забезпечення — персональними ЕОМ і

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

мікрокомп'ютерами, зв'язаними локальною обчислювальною мережею. Крім диспетчерських функцій. SCADA виконує роль інструментальної системи розробки програмного забезпечення для промислових систем комп'ютерної автоматизації, тобто роль специфічної CASE-системи. Для систем АСУТП характерне використання програмувальних контролерів (ПЛК або PLC — Programmed Logic Controller). — комп'ютерів, убудованих у технологічне устаткування.

Функції SCADA:

- збір первинної інформації від датчиків:
- збереження, обробка і візуалізація даних:
- керування і реєстрація аварійних сигналів:
- зв'язок з корпоративною інформаційною мережею:
- автоматизована розробка прикладного ПО.

До розробки програм для програмувальних контролерів звичайно залучаються не професійні програмісти, а заводські технологи. Поетом}' мови програмування повинні бути досить простими, звичайно побудованими на візуальних зображеннях ситуацій. Наприклад, використовуються різні схемні мови. Ряд мов стандартизований і представлений у міжнародному стандарті ІЕС 1131-3.

На рівні керування технологічним устаткуванням в АСУТП виконуються запуск, тестування, вимикання верстатів, сигналізація про несправності, вироблення керуючих впливів для робочих органів програмно керованого устаткування (NC — Numerical Control). Для цього в складі технологічного устаткування використовуються системи керування на базі убудованих контролерів.

Автоматизована система виробничого планування (АСВП).

Автоматизована система підтримки прийняття інженерних рішень (САПР).

Процес прийняття рішень.

ТПР, а отже, і СППР має на увазі під собою використання процесу, представленого на рис.1.



Рис.1. Процес ухвалення рішення.

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

Ми починаємо з якоїсь існуючої проблеми рішення, з яким зштовхнулася людина, якому необхідно прийняти це рішення. Наша увага спрямована на те, щоб застосувати послідовність видимих кроків для внесення ясності, стосовно до ситуації, коли людина, якій необхідно прийняти рішення, змогла почати рекомендовані дії.

Перший крок (постановка задачі) – формальна модель тієї реальної ситуації, з яким зштовхнеться ЛПР. Ми назвемо це формальне представлення проблеми "базою об'єктивних і суб'єктивних моделей". В основі рішення лежить процес, спрямований на виділення альтернативних підходів до рішення проблеми, що логічно погодяться з базою знань (БЗ) і звідси рекомендуються для подальшого аналізу. Потім нам має бути удосконалити оцінку аналізів для того, щоб досягти інтуїції, чому запропонований альтернативний підхід не тільки є логічно правильним, але і виглядає переконливим, що людина буде діяти в повній відповідності з ним. Оцінка може містити деякі варіанти аналізів з тим, що оцінка дійсно відповідає змістові проблеми. У якийсь момент, оцінка покаже, що запропонований альтернативний підхід настільки вірний, що для ЛПР нема рації продовжувати аналіз.

Метою процесу є синтез основи рішення. Малюнок 2 більш детально демонструє роль "базис об'єктивних і суб'єктивних моделей". Основа містить у собі три частини: альтернативний підхід, що вибрав ЛПР; доступна інформація і переваги, що віддаються ЛПР.



Рис.2. Виявлення й оцінка БД.

Концептуальні моделі.

Аналіз існуючих точок зору на специфічні відмінності СППР від інших типів автоматизованих систем у системах проектування дозволяє виділити як основи класифікації СППР наступні найбільш істотні ознаки:

- концептуальні моделі;
- розв'язувані задачі;
- засоби, що забезпечують;
- області застосування.

У рамках інформаційного підходу СППР у промислових САПР і САИТ відносять до класу автоматизованих інформаційних систем, основне призначення яких – поліпшити

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

діяльність працівників розумової праці в організаціях шляхом застосування інформаційної технології.

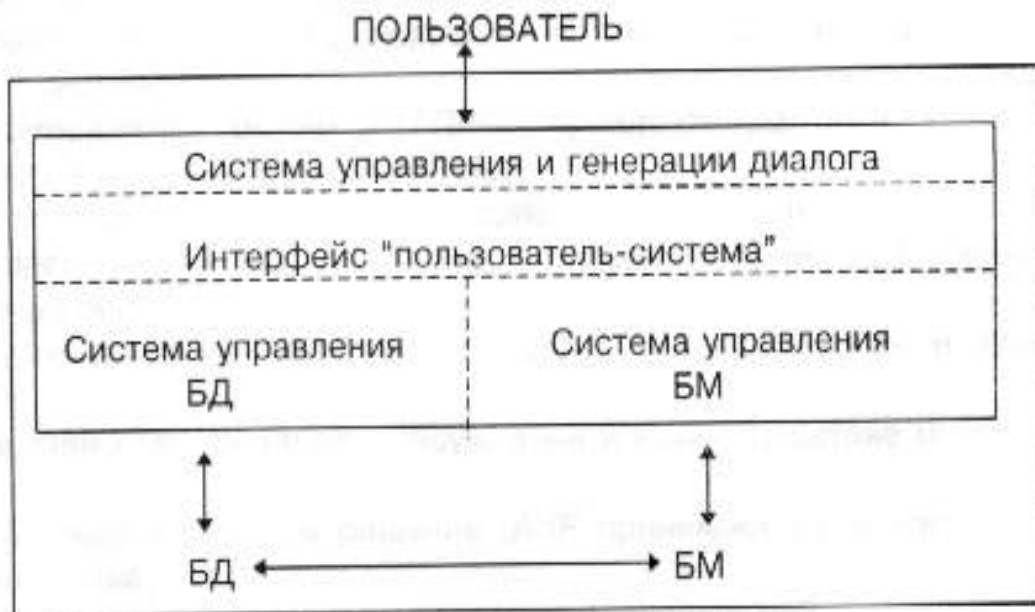


Рис.3.

Концептуальна модель СППР (інформаційний підхід).

Важливою особливістю СППР у промислових САПР і САИТ є їхня здатність формувати моделі для прийняття рішень. Передбачається, що в БМ варто вбудовувати не локальні моделі, а моделі об'єднані з БД. Процедури моделювання повинні забезпечувати гнучкість побудови моделей і легкість керування ними, а система керування – можливість обслуговувати широкий спектр моделей, швидко і легко створювати нові моделі, керувати БД за допомогою функцій керування.

Відмінною рисою СППР у промислових САПР і САИТ, що засновані на знаннях, є, на думку їхніх творців, явне виділення аспекту, що раніше отсутствовали, підтримки рішень: здатності до "розуміння" проблеми, витягти інформацію і підготувати відповідь. Ступінь участі програмних засобів людино-машинної системи в цьому процесі пропонується розглядати як грубу міру ІИ СППР.

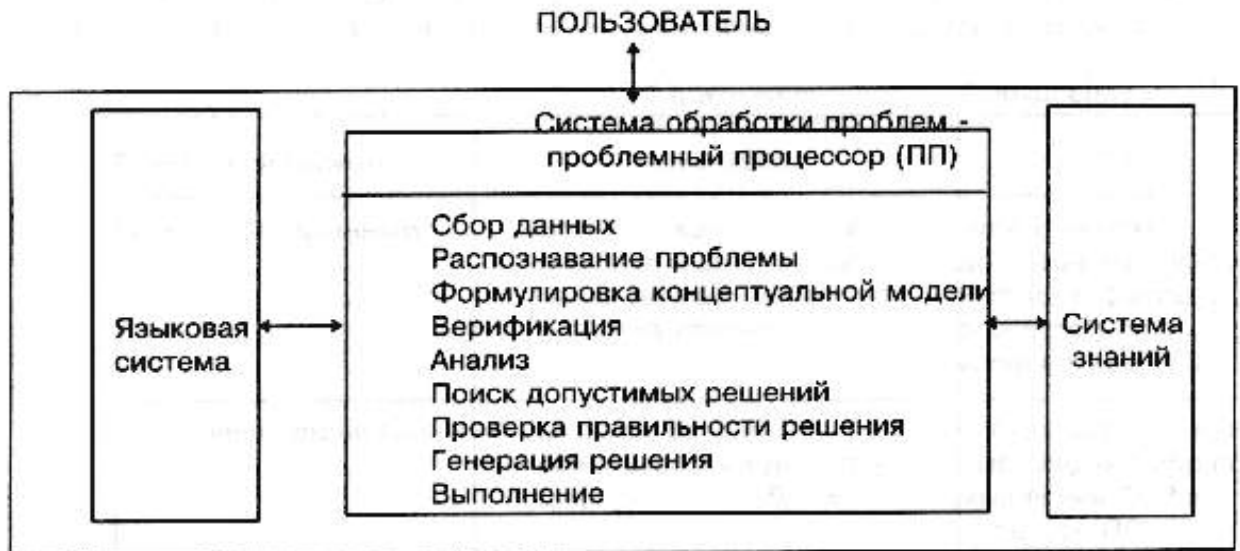


Рис.4.

КП СППР (підхід, заснований на знаннях).

Зросла увага до методів розробки і впровадження СППР у промислових САПР і САИТ обумовило появу інструментального підходу в концептуальних моделях СППР.

Можна виділити три умови систем:

Спеціалізовані САПР (працюють кінцеві користувачі; підтримка рішення окремих прикладних програм (задач) у конкретних ситуаціях);

Сппр-генератори (пакети зв'язаних програмних засобів пошуку і видачі даних);

Сппр-"інструментарий" (вищий рівень технологічності – мови, зроблені ОС, засобу ВВ та інше).

Автоматизована система інженерного документообігу.

Автоматизовані системи діловодства (АСД). Інформаційні технології й автоматизовані системи керування документами і документообігом користуються всі зростаючою увагою серед підприємств і фірм різного профілю, оскільки організація роботи з документами істотно впливає на ефективність виробничих і бізнесів-процесів. Такі системи мають як самостійне значення, так і відіграють важливу роль в інтегрованих автоматизованих системах керування і проектування.

Автоматизовані системи діловодства по своєму призначенню підрозділяють на системи керування документами (СУД), керування документообігом (СДО), керування знаннями

(у сфері діловодства) і інструментальні середовища діловодства. Відповідно до інших критеріїв класифікації системи діловодства підрозділяють на спеціалізовані і комплексні, локальні і розподілені, фактографічні і документографічні (повнотекстові), замовлені і тиражируємі.

Системи керування документами призначені для забезпечення санкціонованого доступу до документів. Характерні функції СУД:

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

— уведення документів, зокрема, за допомогою засобів їхнього автоматичного розпізнавання;

— індексування документів, наприклад, оформлення реєстраційних карток з полями для атрибутів; можливо атрибутивне індексування — до атрибутів відносяться автор документа, дата створення і ключові слова або повнотекстове індексування — в індекс заносять весь текст, але без приводів і закінчень деяких слів.

збереження документів;

пошук потрібних даних, що може бути атрибутивним у фактографічних СУБД або повнотекстовим у випадку слабо структурованих документів:

підтримка групової роботи над документами:

розмежування прав доступу до документів:

контроль і керування версіями документів, що регламентують внесення в них змін:

збір і аналіз статистичних даних по параметрах документів і функціонуванню системи:

підготовка звітів.

Системи керування документообігом служать для керування діловими процесами проходження й обробки документів у відповідних підрозділах і службах організації. Характерні функції СДО:

реєстрація документів при їхньому входженні в систему:

маршрутизація документів, облік їхнього руху (маршрутизація може бути твердої при фіксованих маршрутах або вільної): керування потоками документів забезпечує проходження документів по заданому маршруту з контролем внесення в них резолюцій, керування внесенням змін включає систему пріоритетів, засобу протоколювання змін;

контроль виконання дій, що пропонуються документами,;

защиту інформації при її передачі між пунктами розподіленої системи;

автоматичне повідомлення відповідних осіб про стан документів і директив, що утримуються в них, і рекомендацій:

планування робіт, зв'язаних із проходженням документів.

До систем керування знаннями в області діловодства відносять системи, що виконують функції, характерні для інтелектуальних систем. Приклади таких функцій:

класифікація документів по тим або інших ознаках:

взаємне зв'язування документів, наприклад, за допомогою гіпертексту;

тематичний добір документів:

інтеграція даних, що надходять з різних джерел;

аналітична обробка даних:

моделювання ділових процесів.

Інструментальні середовища в системах діловодства служать для формування систем діловодства, адаптованих до умов конкретних підприємств і фірм. Часто таке формування виробляється шляхом доповнення деякого базового компонента, до складу системи входить відповідна мова розширення.

Крім переліку розв'язуваних задач, виділяють наступні властивості і характеристики систем діловодства:

відкритість, програмні інтерфейси і формати даних для обміну з іншими інформаційними

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

системами:

- мобільність для інсталяції на ведучих платформах;
- модульна побудова, що забезпечує масштабуємість — можливість еволюційного розвитку, адаптируемість, можливість упровадження на підприємствах вроздріб.
- користувальницький інтерфейс;
- швидкодія, часові витрати на виконання задач;
- рівень зашиті інформації;
- відповідність стандартам інформаційних технологій;
- операційні середовища і використовувані СУБД, вимоги до апаратних ресурсів.
- перенос документів по мірі їх старіння на більш дешеві носії.

У великих АСД передбачається розподілене збереження з доступом у режимах як off-line, так і on-line. У першому випадку користувач формує запит у виді сукупності ключових слів і направляє його засобами електронної пошти (E-mail). СДО видає список релевантних документів, користувач вибирає зі списку потрібні документи і посилає вторинний більш конкретний запит, одержуючи по E-mail запитані документи. В другому випадку використовується зв'язок у реальному часі, документ викликається на екран комп'ютера і користувач може безпосередньо його переглядати і редагувати.

Сучасні корпоративні системи діловодства є розподіленими, що мають архітектуру клієнт-сервер. На серверній стороні знаходять застосування сервери баз даних, повнотекстових документів, електронної пошти, додатків. SQL- і Web-сервери. На клієнтській стороні можуть виділятися робочі місця користувачів, адміністратора і розроблювачів баз даних. інформаційно-пошукових систем, форм документів і т.п. Зокрема, застосовуються трьохзвенні розподілені системи.

До широко відомих систем документообігу і діловодства відносяться Lotus Notes, Docs Open, і ін. Переважно використовуваної ОС є Windows XT.

Концепція автоматизованого інтегрованого виробництва

Автоматизоване проектування виробів закінчується виготовленням конструкторської документації і керуючих програм на машинних носіях. На завершальних етапах проектування вносяться технологічні доповнення і корекції. Далі виготовляється спробний зразок. Після його аналізу здійснюється атестація проекту. Це забезпечується на основі введення автоматизованих виробничих ліній до складу технологічного комплексу.

Атестовані машинні носії з керуючими програмами надалі копіюються. На їхній основі виконується перебудова виконавчого устаткування на виготовлення іншого виробу. Відзначене є однією з основних умов реалізації гнучкого автоматизованого виробництва.

Метою АІВ є застосування ефективних методів проектування виробництва в єдиній системі з метою забезпечення економічного благополуччя підприємства. Технологія АІВ базується на повному електронному описі виробів.

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

Стратегія АІВ включає:

- Маркетинг
- Концептуальне і робоче проектування
- Технологічну підготовку виробництва
- Експлуатацію об'єкта
- Його супровід

Прикладом сучасної системи, що реалізує стратегію АІВ, є система CADD' 5. Інтегрально-інструментальне середовище системи включає 85 окремих продуктів функціонально охоплюють:

- Ескізне і робоче проектування
- Синтез геометричних моделей
- Інженерний аналіз
- Розробку документації
- Підготовку виробництва

Працюючи в середовищі параметричного конструювання системи CADD' 5, користувач визначає цільову функцію, указує змінювані параметри і задає єднальні умови. Далі здійснюється настроювання системи і запускається підсистема «Design optimizer», що послідовно переглядає різні варіанти конструкції, наближаючи до оптимального рішення.

Виділяють шість основних функцій АІВ:

Дослідження, розробка і проектування

а) *Базисне проектування* (складова частина – етап *АВАН* проектування і перед-проектні дослідження);

б) Прогностичну оцінку й оптимізацію проектних параметрів;

в) Контроль і адміністрування процесу проектування в термінах інформаційних систем.

Ця функція керує даними:

- дані геометричного моделювання;
- цифровий контроль даних;
- аналіз коректності даних;
- інспекція технічного завдання;
- адміністрування даних відповідно до найбільш ефективних маршрутів проектування).

Ця функція реалізується через систему автоматизації інженерної праці (САПР), а також безпосередньо через модулі *САПР*.

Найбільше успішно ця функція реалізована в системі *PRO\ENGINEER*, що є *CAD/CAM/CAE* системи третього покоління. Система забезпечує роботу проектувальника, з єдиною математичною моделлю об'єкта, а не з набором розрізнених математичних моделей.

Узагальнена математична модель об'єкта може бути представлена у виді багаторівневою, конвеєрною, мережною математичною моделлю, де рівні моделі відбивають рівні проектування (етапи) виробу, конвеєрність моделі відбиває механізми передачі даних і їхній

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

спектр між етапами (не обов'язково сусідніми), і мережна структура моделі відбиває параметричні зв'язки між окремими характеристиками проектованого виробу. Многоуровневості моделі відбиває також процес послідовної деталізації об'єкта в ході проектування.

Ядром усіх сучасних технологій проектування (спадне, наскрізне, рівнобіжне, CALS) є багаторівнева математична модель проектування (конвеєрна, мережна, узагальнена).

Планування виробництва

Містить у собі всі рівні планування, у тому числі:

- а) Довгострокове планування
- б) Динамічне планування, проектування і виробництво
- в) Планування проектування
- г) Планування виробництва
- д) Планування збуту виробу

Організація виробничих процедур

Включає процедури по виробництву, контролеві і збереженню готової продукції

Продаж і маркетинг

Містить у собі:

- а) Установлення планів по виробництву і продажах
- б) Розробку програм по маркетингу
- в) Забезпечення роботи допоміжних служб

Фінансове керування процесами проектування і виробництва, розподіл і облік витрат

Забезпечує постійну випереджальну оцінку витрат при процесах проектування і виробництва

Система адміністрування

Містить у собі всі аспекти керування

- а) Персоналом
- б) Офісом (керування підприємством, власністю, відносинами з громадськістю)

Усі перераховані вище функції будуються на принципі випереджального планування процесів проектування і виробництва виробу.

Керування загальним потоком інформації в компанії, що реалізує технологію АИП, неможливо без елементів підтримки, до яких відносять:

Апаратне забезпечення і програмний контроль

Забезпечення інформаційного зв'язку комп'ютерів

Керування даними

Загальне керування системою

Прикладне програмне забезпечення

Систему підтримки прийняття рішень

Систему керування виробничими функціями

Усі ці сім елементів складають інфраструктуру АИП. У структуру АИП також входять системи реалізації витрат, що включають:

66 Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Тема № 3. САПР в автоматизованому проектуванні та автоматизованому виробництві СОС.

1) Підсистему бухгалтерського обліку

2) Підсистему контролю витрат

Визначає термінові зміни і доповнення, які необхідно внести в проект, а також процес виробництва і випуску виробу.

3) Підсистему планування витрат

Визначає на етапах розробки технічного завдання й *АВАН* проектування величину приблизних витрат на виробництво виробу.

4. Основні поняття та визначення автоматизованого проектування

Об'єкти проектування та об'єкти автоматизації

Методи проектування СОС

Методи проектування СОС можна розділити на:

Прямі аналітичні методи синтезу (розроблені для ряду простих типових механізмів);

Евристичні методи проектування - рішення завдань проектування на рівні винаходів (наприклад, алгоритм рішення винахідницьких завдань [8]);

Синтез методами аналізу - перебір можливих рішень по певній стратегії (наприклад, за допомогою генератора випадкових чисел - метод Монте-Карло) із проведенням порівняльного аналізу по сукупності якісних і експлуатаційних показників (часто використовуються методи оптимізації - мінімізація сформульованої розроблювачем цільової функції, що визначає сукупність якісних характеристик виробу);

Системи автоматизованого проектування або САПР - комп'ютерне програмне середовище моделює об'єкт проектування й визначає його якісні показники, після ухвалення рішення - вибору проектувальником параметрів об'єкта, система в автоматизованому режимі видає проектну документацію.

Інші методи проектування [9, 10, 11].

Проектування виробів радіоелектронної апаратури являє собою багатоетапний (ітеративний) процес. У ході проектування послідовно уточнюється й деталізується опис майбутнього виробу. Цей процес припускає наявність багатьох рівнів опису. На рис.1 зображений процес проектування у виді сукупності основних етапів і переходів між ними, показані основні види документації, одержувані при виконанні етапів.

Наприклад, ескізний проект є результатом ескізного проектування. З іншого боку, ескізний проект служить вихідним документом для технічного проектування.

Переходи від одних етапів проектування до інших у напрямку зверху вниз природні і відповідають нормальному ходові. Переходи в протилежних напрямках виникають, коли на наступних стадіях проектування виявляється неможливість практичної реалізації рішень, прийнятих на попередніх етапах. Це змушує проектувальників переглядати раніше прийняті рішення. Іноді помилки виявляються на етапах виготовлення серійної продукції або навіть у ході експлуатації.

Послідовність проходження етапів розробки виробу, мети і задачі, що коштують перед проектувальниками на окремих етапах, склад проектної документації і вимоги до неї регламентовані відповідними ДСТ.

Коротко охарактеризуємо основні етапи проектування...

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

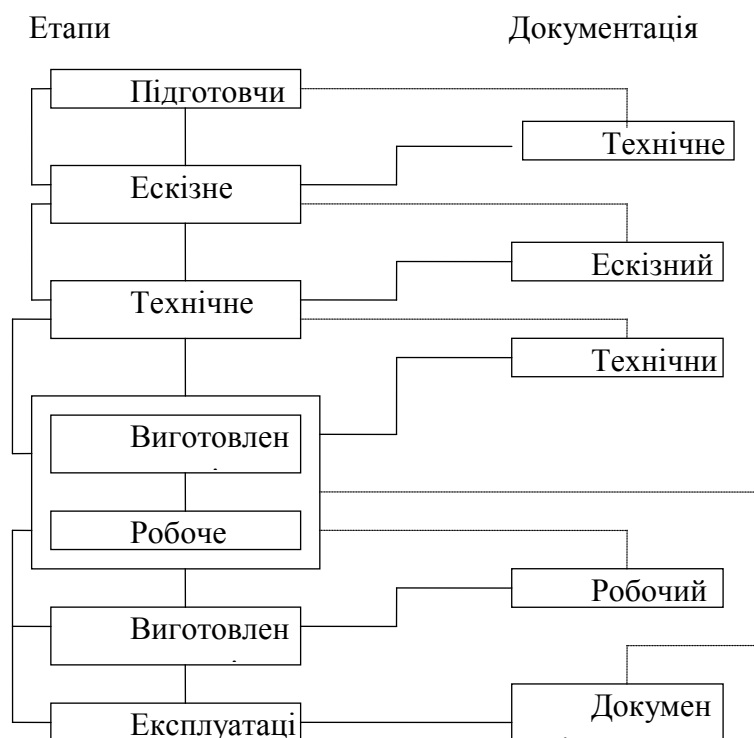


Рис.1. Етапи проектування і документація, що випускається

В загальному випадку, основні етапи процесу проектування можуть включати:

- Усвідомлення суспільної потреби в розроблювальному виробі
- Технічне завдання на проектування (первинний опис)
- Аналіз існуючих технічних рішень
- Розробка функціональної схеми
- Розробка структурної схеми
- Метричний синтез механізму (синтез кінематичної схеми)
- Статичний силовий розрахунок
- Ескізний проект
- Розрахунок і конструювання деталей СОС (міцність, балансування, вітрозахист, тощо.)
- Технічний проект
- Робочий проект (розробка робочих креслень деталей, технології виготовлення й зборки)
- Виготовлення дослідних зразків
- Випробування дослідних зразків
- Технологічна підготовка серійного виробництва
- Серійне виробництво виробу

5. Основні поняття та визначення в проектуванні

Конструкторська ієрархія СОС

Блочно-ієрархічний підхід до процесу проектування

При розробці сучасної технології проектування СОС використовується блочно-ієрархічний підхід, при якому ОП розділяється на ієрархічні рівні. На найвищому рівні застосовують найменш детальне представлення, яке відображає тільки загальні риси і особливості системи. На наступних рівнях степінь деталізації зростає, але система розглядається не в цілому, а окремими блоками. Поділ на блоки повинен бути таким, щоб документація на окремий блок будь-якого рівня могла легко сприйматись одним проектувальником.

Таким чином, при блочно-ієрархічному підході до проектування СОС, складна задача великої розмірності розбивається на послідовні групи задач малої розмірності. Напр., конструкторська ієрархія РЕЗ:

- модуль (блок) 4-го рівня – система;
- модуль (блок) 3-го рівня – стійка, шафа;
- модуль (блок) 2-го рівня – панель, блок;
- модуль (блок) 1-го рівня – ДП, Тези, суб-блоки;
- модуль (блок) 0-го рівня – R, C, D, T, IC\BIC\NBIC, MC, тощо.

Компоненти об'єкта, розглянуті як елементи на деякому рівні з номером k , описуються як підсистеми на сусідньому рівні з номером $k+1$.

На верхньому ієрархічному рівні розглядається весь складний об'єкт, як сукупність взаємодіючих підсистем. При цьому, опис кожної підсистеми не повинен бути занадто детальним, так як це приводить до громіздких описів і неможливості рішення виникаючих проектних задач. На наступному ієрархічному рівні підсистеми розглядаються як окремі системи, що в свою чергу, складаються з деяких частин, які мають більш деталізовані описи. Даний ієрархічний рівень є рівнем підсистем. Процес декомпозиції описів і поблочного їх розгляду із зростанням деталізації можна продовжити до отримання опису блоків, що складаються з базових елементів.

Отже, поділ описів об'єктів, що проектуються на ієрархічні рівні по степені детальності відображення властивостей об'єктів складає сутність “блочно-ієрархічного” підходу до проектування. Відповідно, можливим є поділ проектування, як процесу, на групи проектних процедур.

Групи проектних процедур також утворюють ієрархічні рівні проектування. Напр., ієрархічні рівні конструкторського проектування ЕОМ і РЕА пов'язані з розробкою конструктивів: шаф, рам, панелей, типових елементів заміни (ТЕЗ), тощо.

Ієрархічні рівні технологічного проектування виділяють у відповідності з групами задач проектування принципів схем технологічних процесів, маршрутної і поопераційної

Рівні абстрагування й аспекти описів об'єктів проектування

Більшість видів електронної техніки і радіоелектронної апаратури, а також великі і над великі інтегральні схеми відносяться до складних систем.

Дамо визначення складної системи.

СКЛАДНА СИСТЕМА - система, що володіє, принаймні, одним з перерахованих ознак:

а) допускає розбивку на підсистеми, вивчення кожної з яких, з урахуванням впливу інших підсистем у рамках поставленої задачі, має змістовний характер;

б) функціонує в умовах істотної невизначеності і вплив середовища на неї обумовлює випадковий характер зміни її параметрів або структури;

в) здійснює цілеспрямований вибір свого поведіння.

Процес їхнього проектування характеризується високою розмірністю розв'язуваних задач, наявністю великого числа можливих варіантів, необхідністю обліку різноманітних факторів.

В основі проектування складних систем блочно-ієрархічний підхід. Сутність блочно-ієрархічного підходу полягає в зменшенні складності розв'язуваної проектною задачі. Це здійснюється за рахунок виділення ряду рівнів абстрагування (ієрархічних розрізняються ступенем деталізації представлень про об'єкт.

Аспекти та ієрархічні рівні проектування.

Інженерні представлення про складні технічні об'єкти чи системи (СОС) в процесі їх проектування поділяються на аспекти та ієрархічні рівні.

Аспекти характеризують ту чи іншу групу властивостей ОП, напр.:

Щодо аспектів опису об'єкта.

Аспекти можуть бути:

- функціональні,
- конструкторські,
- технологічні.

В деяких випадках доцільно поділити опис ОП на більшу кількість аспектів. Напр., при проектуванні обчислювальних систем, окремий аспект складає опис алгоритмів і програм; при проектуванні електромеханічних периферійних пристроїв виділяють аспекти електронний і механічний.

Функціональне проектування складних систем найчастіше є низхідним, а конструкторське – висхідним.

Представлення про ОП в середині кожного аспекту необхідно розділити на ієрархічні

Лекція 5. Основні поняття та визначення АП СОС

рівні (рівні абстрагування).

На верхньому ієрархічному рівні розглядається весь складний об'єкт, як сукупність взаємодіючих підсистем. При цьому, опис кожної підсистеми не повинен бути занадто детальним, так як це приводить до громіздких описів і неможливості рішення виникаючих проектних задач. На наступному ієрархічному рівні підсистеми розглядаються як окремі системи, що в свою чергу, складаються з деяких частин, які мають більш деталізовані описи. Даний ієрархічний рівень є рівнем підсистем. Процес декомпозиції описів і поблочного їх розгляду із зростанням деталізації можна продовжити до отримання опису блоків, що складаються з базових елементів.

Отже, поділ описів об'єктів, що проектуються на ієрархічні рівні по степені детальності відображення властивостей об'єктів складає сутність "блочно-ієрархічного" підходу до проектування. Відповідно, можливим є поділ проектування, як процесу, на групи проектних процедур.

Групи проектних процедур також утворюють ієрархічні рівні проектування. Напр., ієрархічні рівні конструкторського проектування ЕОМ і РЕА пов'язані з розробкою конструктивів: шаф, рам, панелей, типових елементів заміни (ТЕЗ), тощо.

Ієрархічні рівні технологічного проектування виділяють у відповідності з групами задач проектування принципів схем технологічних процесів, маршрутною і поопераційною технологією.

Функціональне проектування

Функціональний А. відображає фізичні і (або) інформаційні процеси, що протікають в об'єкті при його функціонуванні. Ф.А. проектування пов'язаний з формуванням і дослідженням структурних, функціональних і принципів схем. Загалом, ФУНКЦІОНАЛЬНИМ називають проектування, пов'язане з рішенням групи задач, що відносяться до функціонального аспекту. При ФП отримують і перетворюють структурні, функціональні і принципів схеми.

Функціональні аспекти можна розділити на:

- системний,
- функціонально-логічний,
- схемотехнічний,
- компонентний.

На системному рівні як системи виділяють комплекси. Прикладами комплексів можуть бути ЕОМ, радіолокаційна станція. Як елементи виділяють блоки (пристрою) апаратури процесор, модем, передавач.

На функціонально-логічному рівні ці блоки розглядають як системи, що складаються з елементів. Елементами є функціональні вузли - лічильники, дешифратори, окремі тригери, вентиля, підсилювачі, модулятори й ін.

На схемотехнічному рівні функціональні вузли описуються як системи, що складаються з елементів радіоелектронних схем - транзисторів, конденсаторів, резисторів і ін.

На компонентному рівні розглядаються процеси, що мають місце в схемних компонентах.

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

Конструкторське проектування

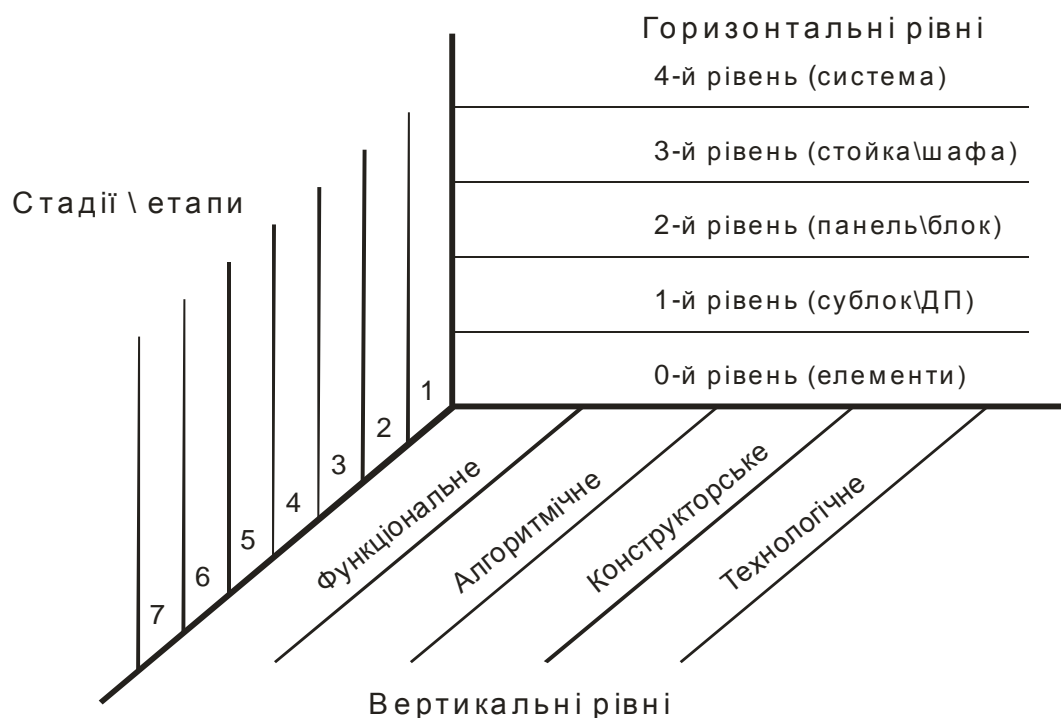
Конструкторський А характеризує структуру розташування в просторі і форму складових частин ОП.

Конструкторському аспекту властива своя ієрархія компонентів. Вона включає різні рівні опису рам, стійок, панелей, типових елементів заміни, дискретних компонентів і мікросхем, топологічних фрагментів функціональних осередків і окремих компонентів у кристалах інтегральних мікросхем.

Технологічне проектування

Технологічний А – технологічність, можливості і способи виготовлення СОС в заданих умовах.

Видлення ієрархічних рівнів, аспектів та стадій технології проектування дозволяє представити їх взаємозв'язок за допомогою наступної схеми:



До *горизонтальних рівнів* проектування попадають однотипні задачі з строго визначеними математичними моделями та апаратом їх аналізу. Основною передумовою появи горизонтальних рівнів проектування є використання блочно-ієрархічного підходу.

Вертикальні рівні формуються на основі властивостей ОП і включають, так звані – аспекти проектування.

Лекція 5. Основні поняття та визначення АП СОС

Аналіз вертикальних рівнів.

Функціональне проектування орієнтоване на синтез та аналіз функціональних, електричних, функціонально-логічних та ін.. схем СОС.

Алгоритмічне проектування (яке присутнє тільки при проектуванні засобів ОТ) відповідає за всі програмні функції, які повинен виконувати СОС.

Конструкторське проектування включає рішення усіх монтажно-комутаційних задач (компоновка -> розміщення -> трасування), які можуть повторюватися на кожному конструкторському рівні ієрархії. Тут також можуть вирішуватись задачі тепло- масо- обміну і механічної стійкості.

Технологічне проектування включає в себе розробку принципової схеми технологічного процесу, маршрутної технології, операційної технології, а також проектування технологічної оснастки.

Структура процесу проектування

Процес проектування поділяється на етапи, процедури, операції, переходи.

Результатом кожної такої «одиниці» повинно бути проектне рішення.

ЕТАП ПРОЕКТУВАННЯ - умовно виділена частина процесу проектування, що складає з однієї або декількох проектних процедур. Звичайно етап включає процедури, що зв'язані з одержанням опису в рамках одного аспекту й одного або декількох рівнів абстрагування. Іноді в процесі проектування виділяють ту або іншу послідовність процедур за назвою "маршрут проектування".

Етапи, у свою чергу, поділяються на процедури й операції.

ПРОЦЕДУРА - формалізована сукупність дій, виконання яких закінчується проектним рішенням.

ПРОЕКТНЕ РІШЕННЯ - проміжний або остаточний опис об'єкта проектування, необхідного і достатнє для розгляду і визначення подальшого напрямку або остаточного проектування.

Ієрархічна структура проектних специфікацій та ієрархічні рівні проектування.

При використанні блочно-ієрархічного підходу до проектування представлення про проектувану систему розчленовують на ієрархічні рівні. На верхньому рівні використовують найменш деталізоване представлення, що відбиває тільки самі загальні риси й особливості проектуваної системи. На наступних рівнях ступінь подробиці опису зростає, при цьому розглядають вже окремі блоки системи, але з урахуванням впливів на кожний з них його сусідів. Такий підхід дозволяє на кожному ієрархічному рівні формулювати задачі прийнятної складності, що піддаються рішенню за допомогою наявних засобів проектування. Розбивка на рівні повинне бути таким, щоб документація на блок будь-якого рівня була обзрима і сприймана одною людиною.

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

Іншими словами, блочно-ієрархічний підхід є декомпозиційний підхід (його можна назвати також діакоптичним), який заснований на розбивці складної задачі великої розмірності на послідовно і (або) паралельно розв'язувані групи задач малої розмірності, що істотно скорочує вимоги до використовуваних обчислювальних ресурсів або час рішення задач.

Можна говорити не тільки про ієрархічні рівні специфікацій, але і про ієрархічні рівні проектування, розуміючи під кожним з них сукупність специфікацій деякого ієрархічного рівня разом з постановками задач, методами одержання описів і рішення виникаючих проектних задач.

Список ієрархічних рівнів у кожному додатку може бути специфічним, але для більшості додатків характерно наступне найбільш велике виділення рівнів:

- системний рівень, на якому вирішують найбільш загальні задачі проектування систем, машин і процесів: результати проектування представляють у виді структурних схем, генеральних планів, схем розміщення устаткування, діаграм потоків даних і т.п.;
- макрорівень, на якому проектують окремі пристрої, вузли машин і приладів; результати представляють у виді функціональних, принципівих і кінематичних схем, складальних креслень і т.п.;
- мікрорівень, на якому проектують окремі деталі і елементи машин і приладів.

У кожному додатку число виділених рівнів і їхніх найменувань можуть бути різними. Так, у радіоелектроніці мікрорівень часто називають компонентним, макрорівень - схемотехнічним. Між схемотехнічним і системним рівнями вводять рівень, названий функціонально-логічним. В обчислювальній техніці системний рівень підрозділяють на рівні проектування ЕОМ (обчислювальних систем) і обчислювальних мереж. У машинобудуванні маються рівні деталей, вузлів, машин, комплексів.

У залежності від послідовності рішення задач ієрархічних рівнів розрізняють спадне, висхідне і змішане проектування (стилі проектування). Послідовність рішення задач від нижніх рівнів до верхнього характеризує висхідне проектування, зворотна послідовність приводить до спадного проектування, у змішаному стилі маються елементи як висхідного, так і спадного проектування. У більшості випадків для складних систем віддають перевагу спадному проектуванню. Відзначимо однак, що при наявності 'ззадалегідь спроектованих складених блоків (пристроїв) можна говорити про змішане проектування.

Невизначеність і нечіткість вихідних даних при спадному проектуванні (тому що ще не спроектовані компоненти) або вихідних вимог при висхідному проектуванні (оскільки ТЗ мається на всю систему, а не на її частині) обумовлюють необхідність прогнозування відсутніх даних з наступним їхнім уточненням, тобто послідовного наближення до остаточного рішення (ітераційність проектування)..

Поряд з декомпозицією описів на ієрархічні рівні застосовують поділ представлень про спроектовані об'єкти на аспекти.

Аспект опису (страти) — опис системи або її частини з деякої обговореної точки зору, обумовленої функціональним, фізичними або іншим типом відносинами між властивостями й

Лекція 5. Основні поняття та визначення АП СОС

елементами.

Розрізняють аспекти функціональний, інформаційний, структурний і поведінковий (процесний). Функціональний опис відносять до функцій системи і найчастіше представляють його функціональними схемами. Інформаційний опис містить у собі основні поняття предметної області (сутності), словесне пояснення або числові значення характеристик (атрибутів) використовуваних об'єктів, а також опис зв'язків між цими поняттями і характеристиками. Інформаційні моделі можна представляти графічно (графи, діаграми сутність-відношення), у виді таблиць або списків. Структурний опис відноситься до морфології системи, характеризує складові частини системи і їх міжзеднання і може бути представлено структурними схемами, а також різного роду конструкторською документацією. Поведінковий опис характеризує процеси функціонування (алгоритми) системи і (або) технологічні процеси створення системи. Іноді аспекти описів зв'язують з підсистемами, функціонування яких засновано на різних фізичних процесах.

Відзначимо, що в загальному випадку виділення страт може бути неоднозначним. Так, крім зазначеного підходу, очевидна доцільність виділення таких аспектів, як функціональне (розробка принципів дії, структурних, функціональних, принципівих схем), конструкторське (визначення форм і просторового розташування компонентів виробів), алгоритмічне (розробка алгоритмів і програмного забезпечення) і технологічне (розробка технологічних процесів) проектування систем. Прикладами страт у випадку ТЗ можуть служити також розглянуті далі види забезпечення автоматизованого проектування.

Поняття стадії (етапу) проектування

Стадії проектування — найбільш великі частини проектування, як процесу, що розвивається в часі. У загальному випадку виділяють стадії науково-дослідних робіт (НИР), ескізного проекту або дослідно-конструкторських робіт (ОКР), технічного, робочого проектів, іспитів досвідчених зразків або досвідчених партій. Стадію НИР іноді називають передпроектним дослідженнями або стадією технічної пропозиції. Очевидно, що в міру переходу від стадії до стадії ступінь подробиці і старанність пророблення проекту зростають, і робочий проект уже повинний бути цілком достатнім для виготовлення досвідчених або серійних зразків.

Близьким до визначення **стадії**, але менш чітко обговореним поняттям, є поняття **етапу** проектування.

Стадії (етапи) проектування підрозділяють на складові частини, називані проектними процедурами. Прикладами проектних процедур можуть служити підготовка деталіровочних креслень, аналіз кінематики, моделювання перехідного процесу, оптимізація параметрів і інші проектні задачі. У свою чергу, проектні процедури можна розчленувати на більш дрібні компоненти, називані проектними операціями, наприклад, при аналізі міцності деталі сітковими методами операціями можуть бути побудова сітки, вибір або розрахунок зовнішніх впливів, власне моделювання полів напруг і деформацій, представлення результатів моделювання в графічній і текстовій формах. Проектування зводиться до виконання деяких послідовностей проектних процедур -маршрутів проектування.

Іноді розробку ТЗ на проектування називають зовнішнім проектуванням, а реалізацію ТЗ — внутрішнім проектуванням.

Зміст технічних завдань на проектування. У ТЗ на проектування об'єкта вказують, принаймні, що впливають дані.

Призначення об'єкта.

Умови експлуатації. Поряд з якісними характеристиками (представленими у вербальній формі) мають бути числові параметри, названі зовнішніми параметрами, для яких зазначені області припустимих значень. Приклади зовнішніх параметрів: температура навколишнього середовища.

зовнішні сили, електричні напруга, навантаження і т.п.

Вимоги до вихідних параметрів, тобто до величин, що характеризують властивості об'єкта, що цікавлять споживача. Ці вимоги виражені у виді умов працездатності

$y_i R T_i$,

де y - J -й вихідний параметр, $R \in$ {дорівнює, менше, більше, більше або дорівнює, менше або дорівнює} — вид відносини: T - норма i -го вихідного параметра. У випадку $R =$ "дорівнює" потрібно задати необхідну точність виконання рівності.

Приклади умов працездатності:

витрата палива на 100 км пробігу автомобіля < 8 л:

коефіцієнт підсилення підсилювача на середніх частотах > 300 :

швидкодія процесора > 40 Мфлопс.

Класифікація моделей і параметрів, використовуваних при автоматизованому проектуванні.

В автоматизованих проектних процедурах замість ще не існуючого проектного об'єкта оперують деяким квазиоб'єктом — моделлю, що відбиває деякі цікаві властивості об'єкта. Модель може бути фізичним об'єктом (макет, стенд) або специфікацією. Серед моделей-специфікацій розрізняють згадані вище функціональні, поведінкові, інформаційні, структурні моделі (опису). Ці моделі називають математичними, якщо вони формалізовані засобами апарата і мови математики.

У свою чергу, математичні моделі можуть бути геометричними, топологічними, динамічними, логічними і т.п., якщо вони відбивають відповідні властивості об'єктів. Поряд з математичними моделями при проектуванні використовують розглянуті нижче функціональні IDEFO-моделі, інформаційні моделі у виді діаграм сутність-відношення, геометричні моделі-креслення. Надалі, якщо немає спеціального застереження, під словом "модель" будемо мати на увазі математичну модель.

Математична функціональна модель у загальному випадку являє собою алгоритм обчислення вектора вихідних параметрів Y при заданих векторах параметрів 'елементів' X і зовнішніх параметрів Q .

Математичні моделі можуть бути символічними і чисельними. При використанні символічних моделей оперують не значеннями величин, а їхніми символічними позначеннями

Лекція 5. Основні поняття та визначення АП СОС

(ідентифікаторами). Чисельні моделі можуть бути аналітичними, тобто їхній можна представити у виді явно виражених залежностей вихідних параметрів Y від параметрів внутрішніх X і зовнішніх Q , або алгоритмічними, у яких зв'язок Y , X і Q заданий неявно у виді алгоритму моделювання. Найважливіший окремий випадок алгоритмічних моделей — імітаційні, вони відображають процеси в системі при наявності зовнішніх впливів на систему. Іншими словами, імітаційна модель — це алгоритмічна поведінкова модель.

Класифікацію математичних моделей виконують також по ряду інших ознак.

Так, у залежності від приналежності до того або іншого ієрархічного рівня виділяють моделі рівнів системного, функціонально-логічного, макrorівня (зосередженого) і мікрорівня (розподіленого).

По характері використовуваного для опису математичного апарата розрізняють моделі лінгвістичні, теоретико-множинні, абстрактно-алгебраїчні, нечіткі, автоматні і т.п.

Наприклад, на системному рівні переважно застосовують моделі систем масового обслуговування і мережі Петрі, на функціонально-логічному рівні — автоматні моделі на основі апарата передатних функцій або кінцевих автоматів, на макrorівні — системи алгебро-диференціальних рівнянь, на мікрорівні — диференціальні рівняння в частинних похідних. Особе місце займають геометричні моделі, використовувані в системах конструювання.

Крім того, уведені поняття повних моделей і макромоделей, моделей статичних і динамічних, детермінованих і стохастичних, аналогових і дискретн, символічн і чисельних.

Повна модель об'єкта на відміну від макромоделі описує не тільки процеси на зовнішніх висновках моделіруємого об'єкта, але і внутрішні для об'єкта процеси.

Статичні моделі описують статичні стани, у них не є присутнім час у якості незалежної перемінної. Динамічні моделі відбивають поведження системи, тобто в них обов'язково використовується час.

Стохастичні і детерміновані моделі розрізняються в залежності від обліку або неврахування випадкових факторів.

В аналогових, моделях фазові перемінні — безперервні величини, у дискретних — дискретні, в окремому випадку дискретні моделі є логічними (булевими), у них стан системи і її елементів описується булевими величинами. У ряді випадків корисне застосування змішаних моделей, у яких одна частина підсистем характеризується аналоговими моделями, інша — логічними.

Інформаційні моделі відносяться до інформаційної страти автоматизованих систем, них використовують насамперед при інфологічному проектуванні баз даних (БД) для опису зв'язків між одиницями інформації.

Найбільші труднощі виникають при створенні моделей слабоструктурованих систем, що характерно насамперед для системного рівня проектування. Тут значна увага приділяється експертним методам. У теорії систем сформульовані загальні рекомендації з підбора експертів при розробці моделі, організації експертизи, по обробці отриманих результатів. Досить загальний підхід до побудови моделей складних слабоструктурованих систем виражений у методиках IDEF.

Звичайно в імітаційних моделях фігурують фазові перемінні. Так, на макrorівні імітаційні моделі являють собою системи алгебро-диференціальних рівнянь

$$J(dV/dt, V, t) = 0, \text{ при } t = 0 \quad V = V_0, (1.1)$$

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

де V — вектор фазових змінних; t — час; V_0 — вектор початкових умов. До прикладів фазових перемінних можна віднести струми і напруги в електричних системах, сили і швидкості — у механічних, тиски і витрати — у гідравлічних.

Вихідні параметри систем можуть бути двох типів. По-перше, це параметри-функціонали, т.е. функціонали залежностей $V(f)$ у випадку використання (1.1). Приклади таких параметрів: амплітуди сигналів, часові затримки, потужності розсіювання і т.п. По-друге, це параметри, що характеризують можливість спроектованого об'єкта працювати при визначених зовнішніх умовах. Ці вихідні параметри є граничними значеннями діапазонів зовнішніх перемінних, у яких зберігається працездатність об'єкта.

Стандартна технологія процесу проектування

Підготовчий етап.

Основна задача - вивчення призначення виробу, умов експлуатації і виробництва, на яких передбачається його виготовлення. Ціль етапу - розробка технічного завдання (ТЗ), у якому утримується інформація про призначення, основних технічних характеристиках, умовах експлуатації, транспортування і збереження.

Ескізне проектування.

Основна задача - визначення можливості розробки виробу у відповідності вимогам ТЗ. При цьому визначають технічну основу виробу (фізичні елементи і деталі), орієнтовану оцінку складу і кількості устаткування, розробляють структуру, визначають технічні характеристики виробу і пристроїв, що входять у його склад.

При цьому може виявитися неможливість побудови виробу, що відповідає вимогам ТЗ. У цьому випадку потрібна коректування ТЗ із наступним його твердженням замовником, або подальша розробка припиняється.

Технічне проектування

Задачі :

- докладна розробка принципу роботи виробу і всіх його складених блоків;
- уточнення технічних характеристик;
- розробка конструкції блоків, вузлів і усього виробу;
- одержання конструкторських характеристик;
- узгодження взаємодії всіх складових частин виробу;
- розробка технології їхнього виготовлення;
- - визначення технології зборки і налагодження, методики і програмні іспити.

У результаті повинне бути підготовлене виробництво дослідного зразка.

Робоче проектування

Основна задача - розробка технологічного оснащення й устаткування для серійного випуску виробу.

Лекція 5. Основні поняття та визначення АП СОС

Впровадження систем автоматизованого проектування (САПР) не змінює суті процесу проектування. Проте, характер діяльності розроблювача з упровадженням САПР істотно міняється, тому що розробка виробу в автоматизованому варіанті припускає погоджену взаємодію оператора й ЕОМ. Це забезпечує істотне підвищення продуктивності праці і підвищення якості проекту.

У процесі автоматизованого проектування на оператора покладаються творчі функції. Як правило, це зв'язано з вибором варіанта рішення, визначення структури, методу розрахунку й ін. Ці функції трудно формалізувати. Тут досвід і талант конструктора, інженера визначають кінцевий результат.

ЕОМ доручають рутинну роботу. Перелічимо її основні види:

- збереження і нагромадження в машинному архіві зведень, необхідних розроблювачеві;
- пошук і видача інформаційних довідок по запитах користувача (типові рішення, характеристики вузлів, рекомендації з застосування, зведення про рівень запасів комплектуючих матеріалів і ін.);
- забезпечення редагування текстової конструкторської документації, створюваної інженером;
- автоматичне креслення графічної документації (креслення деталей, схеми електричні й ін.);
- рішення деяких часток, добре алгоритмізованих задач, що характерні для автоматизованого проектування визначеного класу виробів. Що стосується розробки радіоелектронної апаратури, добре алгоритмізованими задачами є наступні:
 - моделювання поведження того або іншого вузла по описі його принципової електричної схеми при заданому вхідному впливі,
 - трасування з'єднань на етапі конструювання плати друкованого монтажу,
 - розрахунок теплових режимів вузлів апаратури,
 - побудова послідовності обходу точок свердління плати й ін.
 - виділення аспектів опису об'єкта проектування.

6. Маршрут автоматизованого проектування

Типові проектні процедури в САПР

Основні проектні процедури показано на схемі (рис. 1).

Для більшості об'єктів нижчим рівнем проектування можна вважати компонентний. Об'єкт проектування на цьому рівні має структурні й параметричні описи. Таку ж класифікацію можна спостерігати на різних рівнях системи автоматизованого проектування. Будь-яка проектна процедура може бути класифікована на структурну і параметричну.

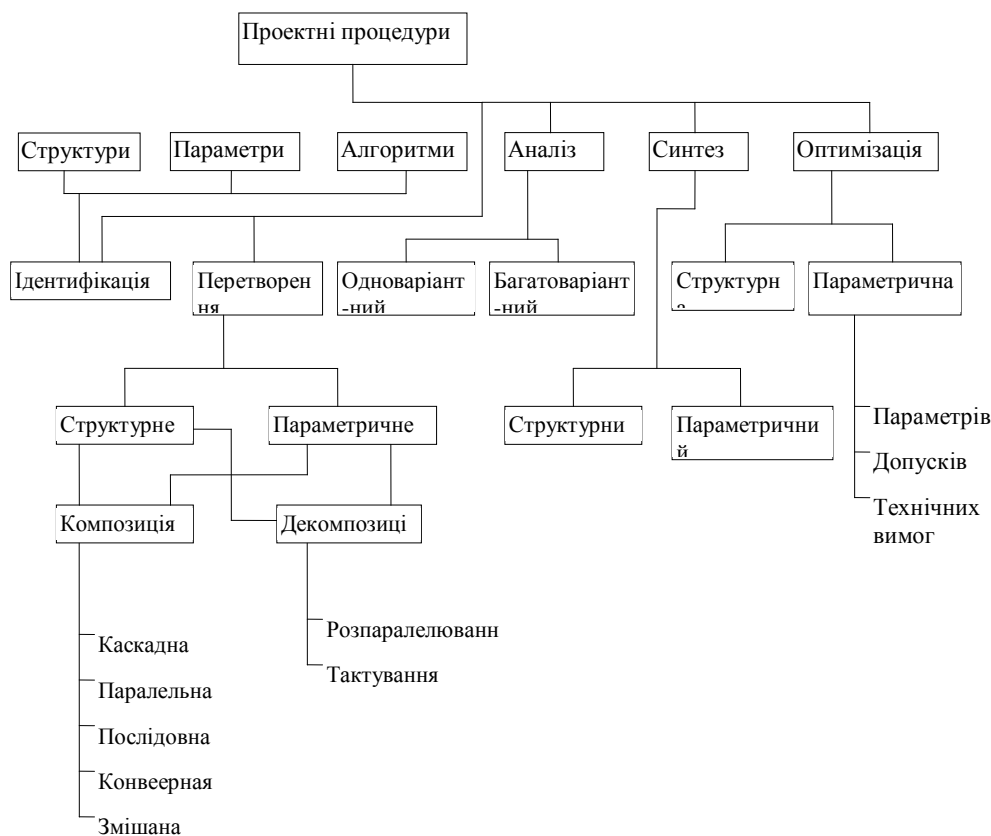


Рис.1. Основні проектні процедури в САПР

Створити проект об'єкта (виробу або процесу) означає вибрати структуру об'єкта, визначити значення всіх його параметрів і представити результати у встановленій формі. Результати (проектна документація) можуть бути виражені у виді креслень, схем, пояснювальних записок, програм для програмно-керованого технологічного устаткування й інших документів на папері або на машинних носіях інформації,

Розробка (або вибір) структури об'єкта є проектна процедура, називана структурним синтезом, а розрахунок (або вибір) значень параметрів елементів X — процедура параметричного синтезу.

Задача структурного синтезу формулюється в системотехніці як задача прийняття рішень (ЗПР). Її суть полягає у визначенні мети, множини можливих рішень і обмежуючих умов. Класифікацію ЗПР здійснюють по ряду ознак. По числу критеріїв розрізняють задачі одно- і

Лекція 7. Технологія паралельного проектування СОС

багатокритеріальні. По ступені невизначеності розрізняють ЗПР детерміновані. ЗПР в умовах ризику — при наявності у формулюванні задачі випадкових параметрів. ЗПР в умовах невизначеності, тобто при неповноті або невірогідності вихідної інформації,

Реальні задачі проектування, як правило, є багатокритеріальними. Одна з основних проблем постановки багатокритеріальних задач — установлення правил переваги варіантів. Способи зведення багатокритеріальних задач до однокритеріальних і наступні пошук рішення вивчаються в дисциплінах, присвячених методам оптимізації і математичному програмуванню.

Наявність випадкових факторів ускладнює рішення ЗПР. Основні підходи до рішення ЗПР в умовах ризику полягають або в рішенні "для найгіршого випадку", або в обліку в цільовій функції математичного чекання і дисперсії вихідних параметрів. У першому випадку задачу вирішують як детерміновану при завищених вимогах до якості рішення, що є головним недоліком підходу. В другому випадку вірогідність результатів рішення набагато вище, але виникають труднощі з оцінкою цільової функції. Застосування методу Монте-Карло у випадку алгоритмічних моделей стає єдиною альтернативою і, отже, для рішення потрібні значні обчислювальні ресурси.

Існують дві групи ЗПР в умовах невизначеності. Одна з них вирішується при наявності протидії розумного супротивника. Такі задачі вивчаються в теорії ігор, для задач проектування в техніку вони не характерні. В другій групі досягненню мети протидію роблять сили природи. Для їхнього рішення корисно використовувати теорію і методи нечітких множин.

Наприклад, при синтезі структури автоматизованої системи постановка задачі повинна включати в якості вихідних даних наступне:

- множина виконуваних системою функцій (іншими словами, множина робіт, кожна з яких може складатися з однієї або більш операцій): можливо, що в цій множині має місце часткова упорядкованість робіт, що може бути представлено у вигляді орієнтованого графа, у якому вершини відповідають роботам, а дуги — відносинам порядку;
- типи припустимих для використання серверів (машин), що виконують функції системи;
- множина зовнішніх джерел і споживачів інформації;
- у багатьох випадках задається також деяка вихідна структура системи у вигляді взаємозалежної сукупності серверів визначених типів: ця структура може розглядатися як узагальнена надлишкова або як варіант першого наближення;
- різного роду обмеження, зокрема, обмеження на витрати матеріальних ресурсів p (або) на час виконання функцій системи.

Задача полягає в синтезі (або корекції) структури, визначенні типів серверів (програмно-апаратних засобів), розподілі функцій по серверах таким чином, щоб досягався екстремум цільової функції при виконанні заданих обмежень.

Конструювання, розробка технологічних процесів, оформлення проектної документації — окремі випадки структурного синтезу.

Задачу параметричного синтезу називають параметричною оптимізацією (або оптимізацією), якщо неї вирішують як задачу математичного програмування

$$\text{extr } F(X), X \in D_X,$$

Лекція 6. Маршрут автоматизованого проектування

де $F(x)$ — цільова функція; X — вектор керованих (називаних також проектними) параметрів:

$Dx = \{X \mid \varphi(X) < 0, \psi(X) = 0\}$ — припустима область: $\varphi(X)$ і $\psi(X)$ — функції-обмеження.

Приклад. Електронний підсилювач: керовані параметри $X =$ (параметри резисторів, конденсаторів, транзисторів): вихідні параметри $Y = (f_v \text{ и } f_n$

— граничні частоти смуги пропускання; DO — коефіцієнт підсилення на середніх частотах; $R_{вх}$ — вхідний опір). Як цільову функцію $F(X)$ можна вибрати параметрів, а умови працездатності інших вихідних параметрів віднести до функцій-ограничителів.

Наступна після синтезу група проектних процедур — процедури аналізу. Ціль аналізу — одержання інформації про характер функціонування і значення вихідних параметрів Y при заданих структурі об'єкта, зведеннях про зовнішні параметри Q і параметрах елементів X . Якщо задані фіксовані значення параметрів X і Q , то має місце процедура одноваріантного аналізу, яка зводиться до рішення рівнянь математичної моделі, наприклад, такий, як модель (1.1), і обчисленню вектора вихідних параметрів Y . Якщо задані статистичні зведення про параметри X і потрібно одержати оцінки числових характеристик розподілів вихідних параметрів (наприклад,

оцінки математичних чекань і дисперсій), то це процедура статистичного аналізу. Якщо потрібно розрахувати матриці абсолютної A і (або) відносної B чутливості, то має місце задача аналізу чутливості.

Елемент A_{ji} матриці A називають абсолютним коефіцієнтом чутливості, він являє собою часткову похідну J -го вихідного параметра y_j , по i -ому параметрові x_i . Іншими словами, A_{ji} — є елементом вектора градієнта J -го вихідного параметра. На практиці зручніше використовувати безрозмірні відносні коефіцієнти чутливості B_{ji} , що характеризують ступінь впливу змін параметрів елементів на зміни вихідних параметрів:

$B_{ji} = A_{ji} x_{iном} / y_{jном}$ — номінальні значення параметрів x_i і y_j відповідно.

У процедурах різноманітного аналізу визначається вплив зовнішніх параметрів, розкиду і нестабільності параметрів елементів на вихідні параметри. Процедури статистичного аналізу й аналізу чутливості — характерні приклади процедур різноманітного аналізу.

Класифікація проектних процедур

Кожна з перерахованих процедур може бути віднесена до одному з трьох типів процедур:

- формальна,
- формалізована,
- евристична.

Дамо кілька визначень.

а) **Формальна процедура** - сукупність дій, що породжує проективне (або множину) проектних рішень без участі проектувальника. У цьому випадку проектант тільки формулює задачу: призначає вхідні дані і вказує критерій проектного рішення.

б) **Формалізована процедура** - процедура, що лише частково формально описує хід отримання проектного рішення у виді якогось алгоритму (іноді чисельного методу), але вихідні дані для неї вимагають перетворення, а умови її протікання, критерії проектних рішень вимагають уточнення не при переході від одного об'єкта проектування до іншого, але і процесі

Лекція 7. Технологія паралельного проектування СОС

її виконання.

Проектант може впливати на умови протікання формалізуемой і евристичної процедур, керуючись тільки лише досвідом, інтуїцією, здоровим глуздом, причому як своїм власним, так і на вибір, і накопиченим у даної САПР.

в) **Евристична процедура** - процедура, що не піддається ніякому формальному опису і не може бути описана ніяким алгоритмом, і за певних умов не забезпечує ухвалення проектного рішення.

Приведемо приклади процедур.

а) Формальні процедури:

- аналіз чутливості моделі проектування, застосовуваної в теорії автоматичного регулювання,

- синтез оптимального фільтра (рішення задачі Винера-Хопфа), коли критерій сформульований у ТЗ,

- пошук безумовного екстремума цільової функції, сформульованої для задачі безпосередньо в ТЗ.

Перераховані процедури дозволяють оперувати зі строгими математичними моделями, є реалізацією конкретно поставленої задачі й інваріантні до критерію проектного рішення.

б) Формалізовані процедури:

- одноваріантний аналіз об'єкта проектування на математичній моделі при відсутності повної інформації про характер вхідних впливів або при нечітко визначеній області адекватності,

- синтез структури об'єкта проектування, що не має аналогів і прототипів, здійснюваний за рахунок використання сполучення різних фізичних принципів і компонент різної фізичної природи. "Сумарний" ефект не є простою сумою окремих ефектів, що виникають від використання того або іншого фізичного принципу,

- умовна оптимізація значень конструктивних параметрів з використанням багатомірної цільової функції (багатокритеріальна некоректно поставлена задача).

Особливою рисою перерахованих формалізованих процедур є їх "екстраполяційний" характер. В усіх зазначених сукупностях дій "передбачується" поведінка об'єкта проектування на основі гіпотетичної інформації при обмеженнях, що засновані на фундаментальних фізичних принципах (закон збереження енергії, принцип старіння, тощо).

Формалізовані процедури займають "проміжне" положення між формальними та евристичними. З одного боку, вони використовують математичні моделі, з іншого боку, характер їхнього використання є екстраполяційним.

Наприклад, параметрична оптимізація, здійснюється зі зміною методу оптимізації. Зміна методу відбувається в результаті аналізу як цільової функції, так і впливу обмежень на характер процесу оптимізації.

Видно, що проектна операція може бути тільки формальною (по визначенню). Тому можна говорити, що формальна проектна процедура завжди складається з кінцевого числа проектних операцій.

Евристична процедура не містить проектних операцій, а формалізуемая складається з несумісних операцій. Їх не можна застосовувати в будь-якій послідовності й у будь-якому

Лекція 6. Маршрут автоматизованого проектування
сполученні, або їхній перетин не можна заздалегідь визначити.

Маршрут проектування в САПР

Маршрут проектування – це послідовність проектних процедур, які приводять до отримання успішного проектного рішення в конкретному процесі проектування СОС.

При проектуванні можливі різні послідовності виконання процедур, стадій (або етапів). В зв'язку з цим, розрізняють два способи проектування (два типи маршрутів):

- висхідне проектування,
- низхідне проектування,
- змішане проектування.

Висхідне проектування (знизу вгору) має місце, якщо проектуються типові об'єкти, призначені для використання як елементи в багатьох об'єктах на більш високих рівнях ієрархії (наприклад, серійні мікросхеми, стандартні осередки матричних великих інтегральних схем).

Низхідне проектування (згори вниз) охоплює ті рівні, на яких проектуються об'єкти, орієнтовані на використання як елементи в одній конкретній системі.

Проектуванню властивий ітераційний характер. При цьому наближення до остаточного варіанта здійснюється шляхом багаторазового виконання однієї і тієї ж послідовності процедур з коректуванням вихідних даних. Ітерації можуть охоплювати різні частини проектування, що включають як кілька операцій, так і кілька етапів.

ПРИКЛАД 1.

- системотехнічне проектування (аналіз тактико-технічних вимог на проєктований комплекс, визначення основних принципів функціонування, розробка структурних схем);
- схемотехнічне проектування (розробка функціональних і принципових схем);
- конструкторське проектування (вибір форми, компоновання і розміщення конструктивів, трасування межсоединений, виготовлення конструкторської документації);
- технологічне проектування (розробка маршрутної й операційної технології, визначення технологічної бази).

ПРИКЛАД 2.

Етапи висхідного проектування БІС:

- приборно-технологічне проектування (вибір базової технології, вибір топології компонентів, розрахунок дифузійного профілю);
- схемотехнічне проектування (синтез принципової електричної схеми, оптимізація параметрів елементів, статистичний аналіз стосовно до типових осередків БІС);
- функціонально-логічне проектування (синтез комбінаційних схем, реалізація пам'яті, синтез контролюючих і діагностичних тестів);
- конструкторсько-топологічне проектування (розміщення елементів, трасування межз'єднань, перевірка відповідності топологічної й електричної схем , розшарування, креслення пошарової технології).

Технічне завдання на об'єкт проектування – це не завжди коректно поставлена задача. Тому при технології спадного проектування, навіть якщо можливо виконання процедури синтезу, проектних рішень завжди може бути більше одного за структурою і значеннями конструктивних параметрів.

Лекція 7. Технологія паралельного проектування СОС

Уточнення ТЗ породжує необхідність виконання процедури аналізу. Помітимо, що при висхідному проектуванні процедура аналізу заданої структури технічного об'єкта проектування стає, як правило, єдиною.

Якщо САПР побудована за принципом генерації проектних рішень, то процедура оптимізації є однією з основних. Навіть у випадку спадного проектування в САПР, що допускає виконання синтезу по ТЗ без генерації варіантів, уточнення значень конструктивних параметрів здійснюється процедурами оптимізації.

У процесі виконання проектних процедур приходиться завжди приводити у відповідність спосіб опису ТЗ, проектного рішення (яке часто є ТЗ для більш низького рівня структури проектування) способів модельного представлення об'єкта проектування.

Перетворенню може бути піддане або ТЗ, або власне модельне представлення.

Таким чином, процедура перетворення описів як структури так і параметрів, є невід'ємною частиною будь-якої САПР.

Багато проектних рішень приймаються в середовищі САПР у формалізованому виді, найчастіше на мета-мові системи. Тому для перетворення опису проектного рішення (а іноді і ТЗ) використовують процедуру ідентифікації проектного рішення (ТЗ).

Множина розглянутих проектних процедур не є замкнутим по номенклатурі і деталізації опису. Однак можна затверджувати, що для розглянутих тут предметних областей воно досить повно деталізує процес проектування.

Останнє твердження ґрунтується на тім, що в зазначених предметних областях існує тією чи іншою мірою адекватне модельне представлення об'єкта проектування у виді деякого оператора:

$$L_a : X \Rightarrow Y ; \quad a \in A,$$

тут X і Y описують множина вхідних і вихідних сигналів .

Дане співвідношення встановлює зв'язок між фазовими перемінними над множиною $\{A\}$ конструктивних параметрів a .

Теоретико-множинний аналіз цього оператора і його відображення для модельного представлення об'єкта проектування показав, що існує рахункова множина його перетворень, а також операцій над ним: декомпозиція і композиція (перетворення) синтез і аналіз. Операції над множиною $\{A\}$ зв'язані з пошуком єдиного a_i , що визначає L_a .

Можливо також або визначення Y при визначених L_a , $\{A\}$ і X , або L_a при заданих Y і X .

Оскільки Y і X найчастіше інваріантні до класу об'єктів проектування в даній САПР, неминуче перетворення описів (представлень) Y , X , $\{A\}$. Формалізм представлення L_a вимагає рішення задачі ідентифікації.

Ітераційний процес проектування.

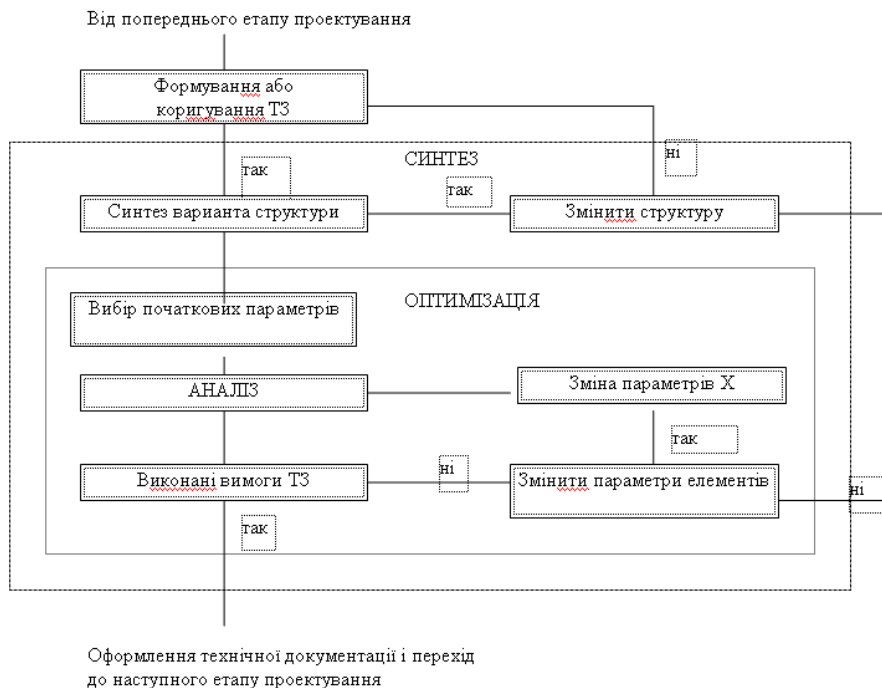


Рис.2. Схема типового маршруту проектування.

Процедура аналізу

Процедура аналізу розуміється як процес представлення вихідних характеристик об'єкта проектування при відомих вхідних впливах або початкових умовах по якому-небудь описі об'єкта проектування (модельному представленню). Одноваріантний аналіз проводиться з метою встановлення відповідності вхідних характеристик вимогам ТЗ. Різноманітний аналіз спрямований на встановлення в явному, наочному виді зв'язку між вхідними даними, характеристиками і конструктивними параметрами.

Процедура синтезу

Рішення задачі синтезу забезпечує одержання опису об'єкта проектування по ТЗ на об'єкт у виді структури і сукупності значень конструктивних параметрів. Параметричний синтез проводиться кінцева множина проектних рішень. Якщо рішення одне, синтезований об'єкт будемо називати "оптимальним по ТЗ". Оптимізація конструктивних параметрів приводить до повного опису об'єкта проектування.

Оптимізація допусків проводиться з метою встановлення сукупності значень кожного конструктивного параметра по заданому теоретично обґрунтованому критерієві. Оптимізація технічних вимог забезпечує призначення оптимальних (у змісті теоретично обґрунтованого критерію) вимог до вихідних характеристик об'єкта проектування.

Процедури перетворення

Процедури перетворення здійснюються над множиначю лінгвістичних і математичн перемінних.

Їхня послідовність утворить запис або ТЗ, або проектного рішення, з метою:

- а) забезпечити представлення ТЗ на метамові конкретної САПР,

Лекція 7. Технологія паралельного проектування СОС

- б) представити у формі, зрозумілої проєктантові, те або інше проєктне рішення, сформульоване на метаязичній системі,
- в) забезпечити сумісність проєктних процедур

Друга група перетворення виділена в окрему процедуру ідентифікації. Якщо перетворення здійснюється над неформальним описом, то воно представимо мовою логіки предикатів першого порядку.

Наприклад, якщо для множини лексем, що утворюють визначену сукупність вимог технічного завдання L^n , і множини критеріїв проєктних рішень K^p ввести предикат $Q(L^n, K^p)$, що позначає факт однозначної відповідності L і K будь-якої лексеми l_i з множини $\{L\}$ однієї і тільки одного формального запису критерію $K_i^p \in \{K\}$, то відповідна формула має вигляд:

$$(\forall l_i^n \in L^n) (\exists K_i^p \in K^p) \Leftrightarrow Q(L^n, K^p).$$

Якщо перетворення не здійснене за одну процедуру, але існують дві сумісні процедури і результат перетворення не залежить від порядку їхнього застосування, то

$$(\forall l_i^n \in L^n, \exists K_i^p \in K^p) \Leftrightarrow Q(L^n, K^p) \Leftrightarrow Q(L^{n-m}, K^p).$$

Якщо як терміни, що входять у формулу, використовуються математичні перемінні, то операція перетворення записується математичною формулою, що реалізується алгоритмічно.

Процедура перетворення, застосовувана до того або іншого проєктного рішення, записаному у виді топології об'єкта проєктування, може здійснюватися або об'єднанням (композицією), або декомпозицією. У свою чергу, композиція проєктних рішень має кілька модифікованих представлень і їхня множина счетно.

Те ж можна сказати щодо декомпозиції проєктного рішення. Тут відзначимо, що процедура композиції характерна для висхідного проєктування, а декомпозиція - для спадних.

Процедура ідентифікації

Процедура ідентифікації здійснюється або над множиною даних (з бази даних) коли ставиться задача перетворення проєктного рішення, або над множиною проєктних рішень, коли ставиться задача занесення його в базу даних.

Зазначені задачі, розв'язувані відповідно до класифіцируемими процедур, як відзначалося, зроблене щодо об'єктів. Вони мають опис у виді ланцюжків формульних залежностей (модельних представлень, реалізованих алгоритмічно) або ланцюжків термов, лексем, лінгвістичних перемінних). Тому можна припустити, що множина перерахованих процедур факторизується по класифікаційній ознаці, зв'язаній зі способом опису і реалізації в середовищі САПР.

Процедури синтезу, аналізу та верифікації в автоматизованому проєктуванні.

Виділяють дві основні задачі в САПР:

I. Задача аналізу. Зв'язана з визначенням функції об'єкта або системи по заданому описі й оцінкою можливих проєктних рішень.

Лекція 6. Маршрут автоматизованого проектування

2. Задача синтезу. Зв'язана зі створенням самого об'єкта і проектною документацією.

Синтез буває:

1. Структурний – одержання структурної схеми об'єкта, формування зведень про склад елементів і способи їхнього з'єднання між собою.

2. Параметричний – визначення числових значень параметрів елементів або систем. Синтез називають оптимізацією, якщо визначаються найкращі в заданому змісті структури (структурна оптимізація) або параметри (параметрична оптимізація). Проведення оптимізації вимагає завдання спектра критеріїв оптимізації і відповідних обмежень. Якщо критерій один, то скалярна оптимізація, якщо трохи – векторна.

Критерії бувають: 1. Часткові. 2. Аддитивні. 3. Мультиплікативні. 4. Мінімаксні.

Для проведення задач аналізу застосовують математичні моделі і мови моделювання (імітаційне моделювання), що дозволяють проводити аналіз об'єкта в динаміку (VHDL і VERILOG).

Класифікація проектних процедур приведена в табл.1.

ТАБЛИЦЯ 1. ПРОЕКТНІ ПРОЦЕДУРИ

АНАЛІЗ		СИНТЕЗ	
Одноваріантний	Багатоваріантний	Параметричний	Структурний
Статики	Чутливості	Розрахунок внутрішніх параметрів	
Динаміки	Статистичний	Оптимізація параметрів	
У частковій області	Розрахунок залежностей вихідних параметрів	Оптимізація допусків	
Стаціонарних режимів	від внутрішніх і зовнішніх параметрів	Оптимізація технічних вимог	
Стійкості			

У процедурах аналізу оцінюються варіанти побудови об'єктів, а в процедурах синтезу - розробляються.

Одноваріантний аналіз полягає у визначенні вектора вихідних параметрів Y при заданих:

- структурі системи,
- значеннях векторів параметрів елементів X ,
- значеннях зовнішніх параметрів Q .

Структура системи задана, якщо задані перелік типів елементів і спосіб їхнього зв'язку

Лекція 7. Технологія паралельного проектування СОС

один з одним у складі системи. За відомою структурою і значенням X і Q можуть бути створені фізична або математична моделі і за результатами дослідження моделі оцінені значення g параметрів вектора Y .

Прийнятність отриманих значень вихідних параметрів з вектора Y визначається шляхом зіставлення їх зі значеннями параметрів з вектора T , зазначених у технічному завданні (ТЗ).

Необхідне по ТЗ співвідношення між значеннями параметрів y_i і t_i , $i=1,n$ називають умовою працездатності по параметрі y_i .

Умови працездатності можуть бути представлені в наступному виді:

$$y_i \leq t_i, \quad (1)$$

$$y_i \geq t_i, \quad (2)$$

$$t_{ni} \leq y_i \leq t_{vi} \quad (3)$$

Задачі аналізу, як правило, полягають у визначенні змін вектора Y при заданих змінах векторів X і Q .

До типових процедур різноманітного аналізу відносяться наступні:

- аналіз чутливості - оцінка впливу внутрішніх і зовнішніх параметрів на вихідні. При цьому здійснюється розрахунок коефіцієнтів чутливості;

- статистичний аналіз - оцінка закону і (або) числових характеристик розподілу вектора Y при заданих статистичних зведеннях про розподіл випадкового вектора X .

При синтезі виділяють процедури параметричного і структурного синтезу. При параметричному синтезі визначаються числові значення параметрів елементів при заданих структурі об'єкта і діапазоні можливої зміни зовнішніх перемінних. Якщо при цьому ставиться задача досягнення екстремуму деякої цільової функції, то виконується процедура оптимізації.

При оптимізації параметрів визначаються номінальні значення внутрішніх параметрів, при оптимізації допусків - додатково допуски на внутрішні параметри, при оптимізації технічних вимог вирішується задача оптимального призначення технічних вимог до вихідних параметрів об'єкта.

У більшості маршрутів проектування процедури синтезу й аналізу знаходяться у взаємозв'язку. На рис. 2 показана схема типового маршруту проектування.

Після формування ТЗ (вихідного опису об'єкта проектування) і вибору (синтезу) первісного варіанта структури і значень параметрів елементів впливає аналіз об'єкта. Якщо при аналізі необхідно установити відповідність синтезованої структури вихідному описові, то аналіз називають верифікацією проекту.

Розрізняють верифікацію параметричну і структурну. При параметричній верифікації установлюється відповідність областей працездатності двох порівнюваних варіантів об'єкта. Прикладом параметричної верифікації є процедура встановлення працездатності типового елемента заміни (Теза). При структурній верифікації перевіряється відповідність структур об'єкта, представлених двома різними описами. Приклад структурної верифікації - встановлення ізоморфізму графів, що описують топологію і принципову електричну схему типового елемента заміни.

Звичайно за результатами аналізу приймається рішення щодо поліпшення первісного
92 Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»
ч.І

Лекція 6. Маршрут автоматизованого проектування

варіанта. Це виконується шляхом зміни числових значень параметрів елементів. Даний процес можна формалізувати і представити як рішення задачі параметричної оптимізації.

Якщо після завершення оптимізації вимоги ТЗ не виконані, то приймається рішення на зміну структури. Після цього здійснюється зазначена послідовність операцій.

Якщо не вдається одержати задовільні результати, ставиться питання щодо корекції ТЗ.

Повний і ретельний аналіз вимагає великих матеріальних і часових витрат. Тому на перших ітераціях у маршруті, показаному на рис.2, виконують спрощений аналіз.

Використання складних моделей, проведення параметричної верифікації і всебічного різноманітного аналізу доцільно лише на завершальних ітераціях.

Для функціонального проектування характерні великі витрати на аналіз. Прикладами подібних задач є такі як визначення складу пристроїв обчислювальної системи і способів їхньої взаємодії або задач розробки принципів електричних схем. Для цих задач звичайно застосовують евристичні способи синтезу структури з перебором невеликого числа варіантів. Основні зусилля затрачаються на виконання різноманітного аналізу й оптимізації.

Якщо вдається організувати наближену оцінку варіантів структури на основі критеріїв, що легко перевіряються, то можливо перегляд великого числа варіантів структури. Це дає можливість формалізувати процес рішення задачі синтезу.

Принципи побудови маршрутів проектування (МП)

Приклади реальних МП

The framework of the top-down design methodology for BioMEMS is illustrated in Fig. 1. [5]

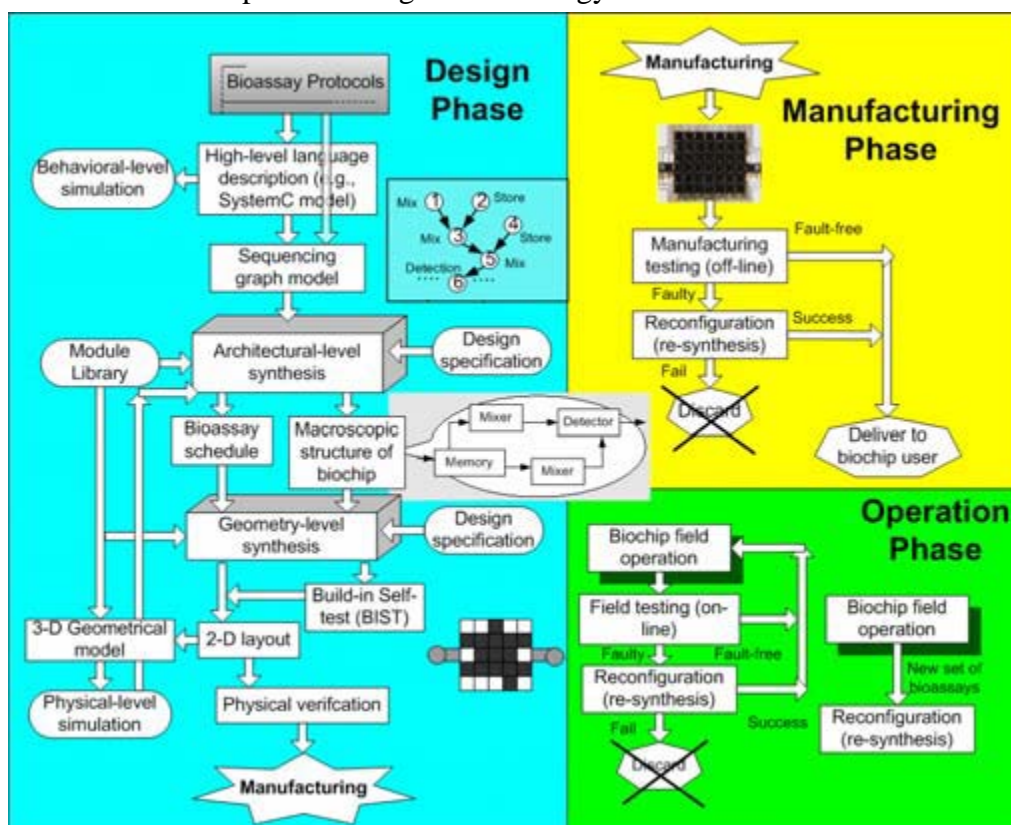


Рис. 1. Overview of top-down design methodology.

Лекція 7. Технологія паралельного проектування СОС

First biochip users, e.g., biochemists, provide the protocol for nano- and micro-scale bioassays. A sequencing graph $G(V, E)$ can directly be applied to describe this assay protocol, where vertex set $V = \{v_i: i = 0, 1, \dots, k\}$ in one-to-one correspondence with the set of assay operations and edge set $E = \{(v_i, v_j): i, j = 0, 1, \dots, k\}$ represents dependencies between assay operations. We can also use a high-level description language such as SystemC to model the protocol, and then derive a sequencing graph model from it. Moreover, this model can be used to perform behavioral-level simulation to verify the assay functionality at the high level [6].

Next, a synthesis tool is applied to generate detailed implementations of digital microfluidic biochips from the sequencing graph model [23]. A microfluidic module library is also provided as an input of the synthesis procedure. This module library, analogous to a standard/custom cell library used in cell-based VLSI design, includes different microfluidic functional modules, such as mixers and storage units. Each module is characterized by its function (mixing, storing, detection, etc.) and parameters such as width, length and operation duration. The microfluidic modules can be characterized through experiments, and their parameters can be stored for use by CAD tools that support large-scale biochip design.

In addition, some design specifications are also given a priori, e.g., an upper limit on the completion time, an upper limit on the size of microfluidic array, and the set of nonreconfigurable resources such as on-chip reservoirs/dispensing ports and integrated optical detectors.

The proposed synthesis tool performs both architectural-level synthesis (e.g., scheduling and resource binding) and geometry-level synthesis (e.g., module placement and routing) [23, 24]. The output of the synthesis tools includes the mapping of assay operation to on-chip resources, a schedule for the assay operations, and a 2-D biochip physical design (e.g., the placement of the modules). The synthesis procedure attempts to find a desirable design point that satisfies the input specifications and also optimizes some figures of merit, such as performance and area. Moreover, since digital microfluidics-based biochips need to be tested adequately not only after fabrication, but also continuously during in-field operation, self-testing has played an important role in the enhancement of biochip yield and reliability. Thus design-for-test (DFT) is also incorporated in the proposed synthesis procedure, whereby a test plan and a set of test hardware (e.g., test droplet sources/sinks and capacitive detection circuits) associated with the synthesized assay operation and biochip physical design are generated [25, 26]. After synthesis, the 2-D physical design of biochip (i.e., module placement and routing) can be coupled with detailed physical information from a module library (associated with some fabrication technology) to obtain a 3-D geometrical model. This model can be used to perform physical-level simulation and design verification at the low level. After physical verification, a digital microfluidics-based biochip design can be sent for manufacturing.

Compared to the full custom and bottom-up design methods, this top-down system-level design methodology not only reduces the design cycle time and time-consuming redesign efforts, but it can also deal with design-for-test (DFT) and design-for-reliability (DFR) issues efficiently.

Методи рішення задач верифікації

Типи проектних процедур

7. Технологія паралельного проектування СОС

Як відомо, в галузі проектування складних систем найперспективнішими на сьогодні є дві технології: **наскрізне** та **паралельне** проектування .

Суть наскрізного проектування полягає в ефективній передачі даних і результатів конкретного поточного етапу проектування відразу на всі наступні етапи. Це потрібно для того, щоб на необхідному рівні якості виконати певний етап проектування. Розробнику часто не вистачає регламентованої інформації від попереднього етапу і необхідна повніша і різноманітніша інформація, яка могла б бути сформульованою на одному з ранніх етапів проектування (не обов'язково на сусідньому). У розробників, які виконують різні етапи проектування, може бути одночасно з першим етапом проектування отримане технічне завдання на інший етап і, таким чином, всі розробники можуть одночасно розпочати продумувати як найуспішніше реалізувати свій етап. Ця технологія базується на модульній побудові САПР, на використанні спільних баз даних і баз знань, і характеризується широкими можливостями моделювання і контролю на всіх етапах проектування. Наскрізні САПР, як правило, є інтегрованими, мають альтернативні алгоритми реалізації окремих проектних процедур.

При паралельному проектуванні інформація про будь-які проміжні або кінцеві характеристики розроблюваного виробу формується і надається всім учасникам робіт, починаючи з найперших етапів проектування. Принциповою відмінністю паралельного проектування (ПП) від наскрізного проектування (хоча паралельне проектування отримало розвиток на основі наскрізного) в тому, що інформація не просто надходить на всі наступні етапи проектування, але і, по суті, ці етапи починають виконуватися одночасно. Особливо перспективним цей підхід є для проектування пристроїв, які мають гетерогенне походження і для їхнього створення потрібно залучати різногалузеві САПР.

Паралельне проектування має важливе значення в новій генерації передових промислових технологій. Паралельне проектування значно прискорює час проектування і виготовлення нових виробів.

Організація життєвого циклу виробу при ПП показана на рис.1.

Найвідоміше визначення паралельного проектування і виготовлення було дане у звіті R-338 Institute for Defense Analyses of America (IDA):

ПП є систематичним методом для інтегрованого, паралельного проектування виробів та пов'язаних із цим інших процесів, включаючи етапи виготовлення та обслуговування. Цей підхід має на меті стимулювати розробників опрацьовувати всі елементи життєвого циклу виробу від ідеї до використання, враховуючи якість, ціну, вимоги користувачів тощо.



Рис. 1. Організація ЖЦВ під час паралельного проектування

Враховуючи вищенаведене визначення і з нашого досвіду в галузі використання ПП за останні роки, *концепцію ПП можна сформулювати так:*

Метою ПП є підвищення якості виробу, зменшення його ціни, скорочення часу виходу на ринок та відповідність вимогам користувачів;

ПП є систематичним робочим циклом. Він підкреслює інтеграцію всього життєвого циклу виробу, включаючи аналіз вимог ринку, опис виробу, дослідницькі роботи, проектування, виготовлення, обслуговування (а також якість, продаж, купівлю, постачання та обслуговування) і утилізацію виробу, а також приділяє особливу увагу паралельним роботам та оптимізації процесів. Проектувальники, які розробляють виріб знизу-вверх, мають усвідомлювати вимоги, що надходять у зворотному напрямку. Особливо на етапі проектування виробу проектувальник має дотримуватися правил проектування для виробництва, оснащення, складальних операцій, тестування, підтримки і ціни. Рекомендується, щоб проектувальник на різних етапах життєвого циклу виробу постійно спілкувався і координував свою роботу з іншими учасниками проектування для уникнення великих обсягів перепроjektування;

ПП є командним процесом, який вимагає не лише кооперації і координації зусиль між різними відділами, а й інтеграції знань від усіх членів колективу розробників;

ПП застосовує передові технології проектування і передові виробничі технології для підтримання цифрового подання виробу та пов'язаних з ним процесів. Це включає в себе системи автоматизованого проектування (САПР), системи технологічної підготовки виробництва (АСТПВ), системи управління (АСУ), системи проектування певних етапів життєвого циклу виробу (DFx) та інші інженерні пакети і інформаційну інтеграцію на різних етапах життєвого циклу виробу;

ПП вимагає наявності висококваліфікованого персоналу, особливо із досвідом і здатністю до командної роботи.

Лекція 6. Маршрут автоматизованого проектування

При ПП фокусується увага на запитах замовника. Логічно та ефективно враховуються фактори виробничих процесів, командної роботи, управління і технології. Спираючись на інтегроване середовище підтримки, оптимізуються потоки інформації, матеріалів та коштів під час проектування виробу та процесів, а також під час виробничої діяльності, управління та обслуговування. Отже, покращуються такі параметри виробу, як час виходу його на ринок, якість виробу, ціна, обслуговування та його вплив на навколишнє середовище.

Технологія ПП розділяється на три стадії (рис.2): формування специфікації вимог (А), концептуальне проектування (В) і деталізоване проектування (С). На першій стадії здійснюється аналіз початкових вимог та обмежень, дається оцінка можливості знаходження проектного рішення, на другій – вибір допустимих (з погляду наступного комплексування) типів проектних рішень (концепцій реалізації елементів моделі предметної галузі), на третій - вибір технічних рішень.

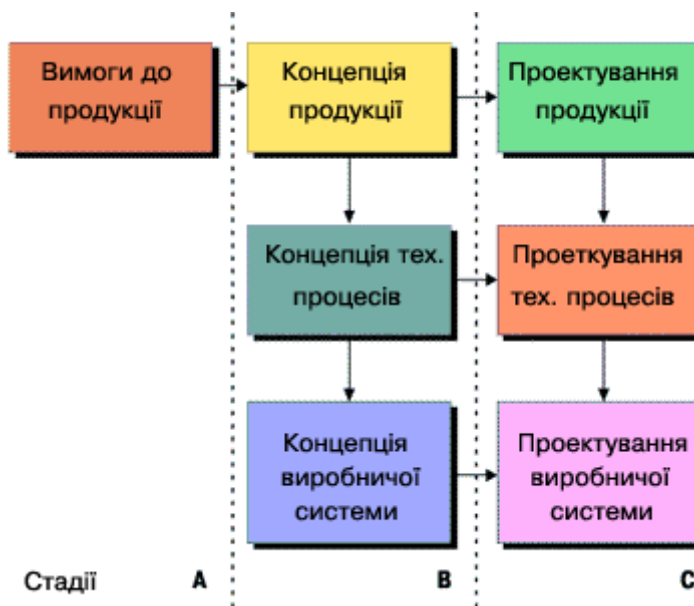


Рис. 2. Основні стадії паралельного проектування виробу

Традиційний процес проектування виробу є переважно лінійним. Його типова схема зображена на рис. 3. У традиційному проектуванні кожний крок проектних робіт реалізується низкою послідовних підзадач. Наступний крок може розпочатися лише тоді, коли завершився попередній. Помилки у проекті переважно виявляють на останніх етапах проектування, і навіть вже на етапі виробництва. Це продовжує виробничий цикл: проектування-виробництво – проектування-модифікація – перевиготовлення, що призводить до збільшення часу виготовлення виробу, негарантованої якості тощо. Така схема не може задовольняти швидкі вимоги сучасного проектування. Для зменшення часу на проектування і виготовлення нового виробу необхідно вдосконалити існуючий стан справ. І тоді, базуючись на дослідницьких роботах, впровадити паралельне проектування, використовуючи операції моделювання процесів.

Схема операцій паралельного проектування та виготовлення виробу і його виробництва показана на рис. 4. Можна побачити, що попередній тривалий цикл робіт (рис. 3) поділений на кілька коротших циклів. За допомогою попередніх схем проектування та перегляду процесів

Лекція 7. Технологія паралельного проектування СОС

проектувальник із суміжної дисципліни може взяти участь у проектуванні виробу швидше, виявити помилки раніше і видалити потенційні проблеми якнайшвидше. Це значно зменшує розміри ризику розробки та виготовлення виробу, вирішуючи проектні завдання на початкових стадіях проектних робіт.

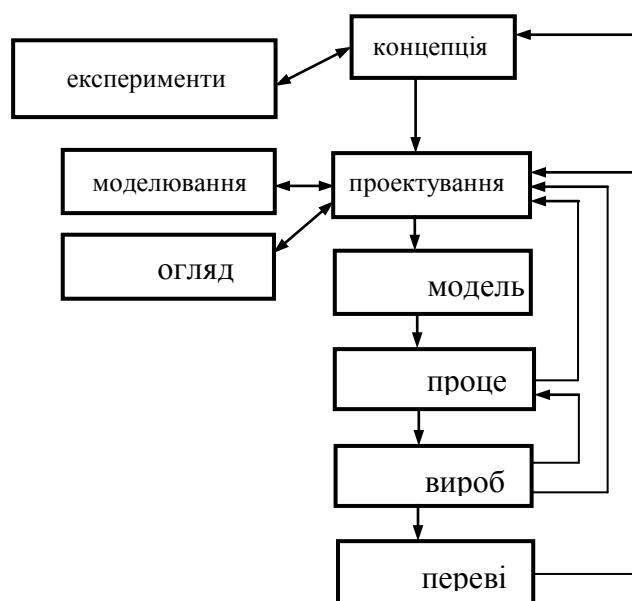


Рис. 3. Послідовний процес проектування та виготовлення виробу

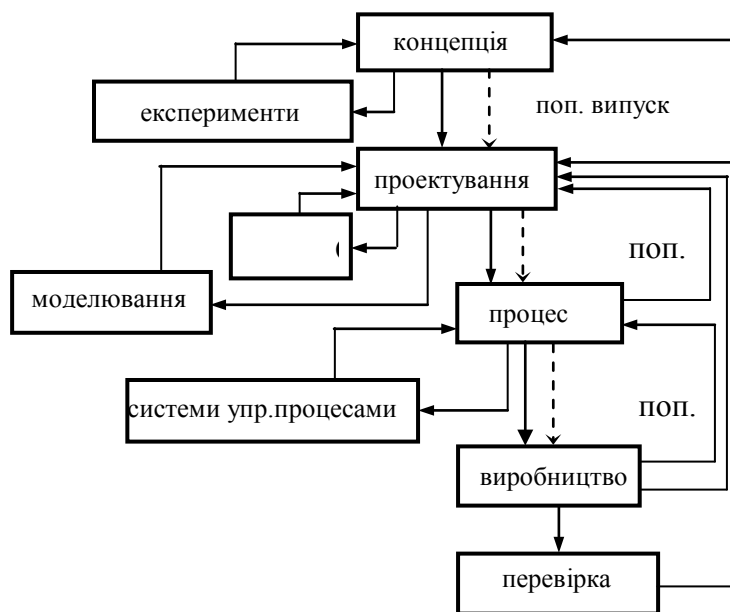


Рис. 4. Паралельний процес проектування виробу

Комп'ютерна система автоматизації проектування САПП повинна в ідеальному випадку виконувати одночасно роль платформи інтеграції програм різного типу, які використовують на різних етапах проектування для автоматизації праці інженера, забезпечувати ефективний пошук та обмін різномітною інформацією, а також бути засобом чіткого планування і спостереження перебігу проектування.

Лекція 6. Маршрут автоматизованого проектування

Система повинна мати можливість оптимізації проектування залежно від доступних людських та інших ресурсів підприємства та від ієрархії поставлених завдань. Звідси випливає необхідність забезпечення обміну інформацією між системою САПП і системами планування і виготовлення продукції. Таку систему можна трактувати як одну з ІКВ (англ. CIM). Місце такої системи у традиційній схемі обміну даними в системі ІКВ показано на рис. 5.

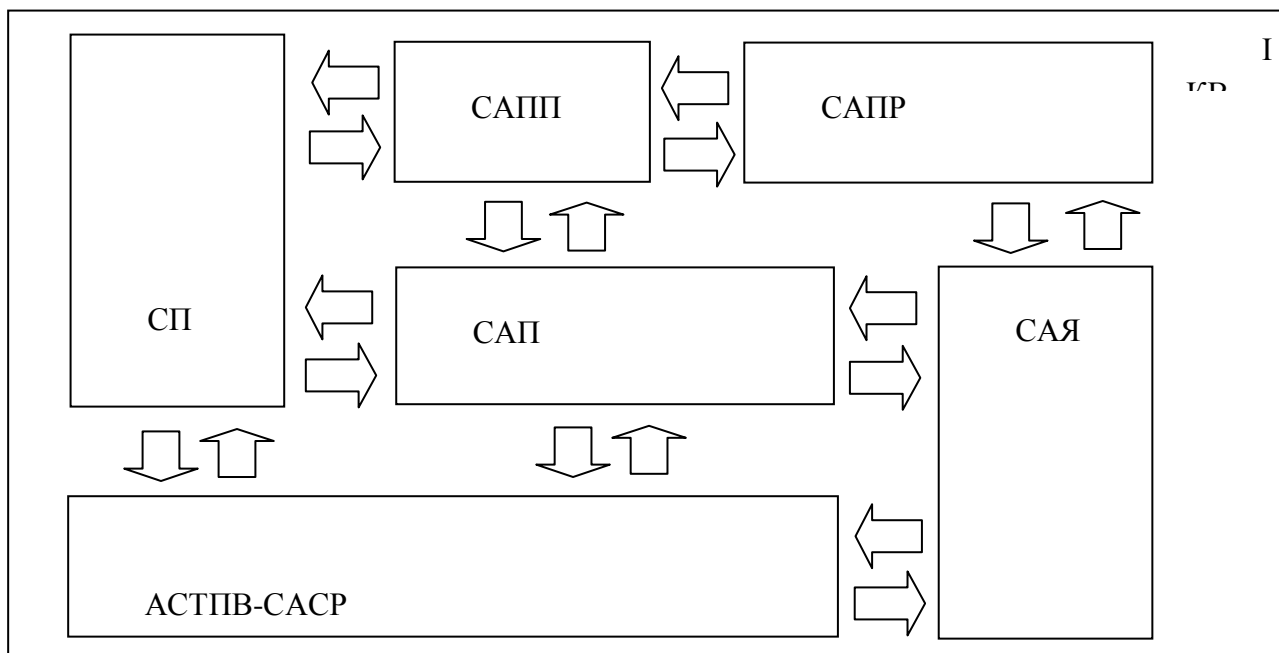


Рис. 5. Взаємозв'язок між підсистемами у системі інтегрованого комп'ютерного виробництва (ІКВ):(САП – Система автоматизованого планування, САСР – Система автоматизації складальних робіт, САЯ – Система автоматизації контролю якості, САТР – Система автоматизації тестувальних робіт, СП – Система планування)

Зважаючи на концепцію паралельного проектування та виготовлення традиційну, схему взаємозв'язків між системами автоматизації у ІКВ у нашій роботі ми зображаємо у вигляді прототипу інтегрованої моделі, що охоплює всі аспекти автоматизації проектування та виготовлення, включаючи технічні та програмні засоби для виконання цих завдань.

Розроблена архітектура системи ПП для всього життєвого циклу виробу зображена на рис. 6. Ми включили до неї чотири підсистеми:

Підсистема управління та дотримання якості виробів, яка аналізує біжучі процеси виготовлення виробу, використовує комп'ютерні технології управління та контролю інтегрованих процесів командної роботи та застосовує метод QFD (Quality-Function Deployment) для підтвердження можливості забезпечення вимог користувача на етапі проектування виробу;

Підсистема проектування базується на інформаційній інтеграції типу ІКВ та технологій САПР/АСТПВ/САПП/САП, також використовує функції системи проектування складальних операцій (СПСО, англ. Design for Assembly - DFA), промисловий стандарт STEP і комп'ютерні засоби для підтримки концептуального дизайну, дизайну структури, дизайну складових та планування процесів. Вона також забезпечує формування цілісної комп'ютерної

Лекція 7. Технологія паралельного проектування СОС

моделі виробу;

Підсистема підтримки діяльності використовується як Система управління даними про виріб (УДВ) для створення платформи для членів колективу і для ширшого кола задіяних осіб, для обміну даними про виріб. Це передбачає формування системної архітектури типу клієнт/сервер та мережу. У такий спосіб географічно віддалені працівники колективу можуть здійснювати спільне проектування виробу за допомогою системи УДВ та системи підтримки групової роботи;

Виробнича система задіює задачі дослідження та розробки і крок за кроком адаптується до конкретного виробництва. Розраховано на виготовлення та складання високоякісних механічних компонентів. Ця підсистема є остаточним виробничим середовищем для ПП. Вона оперує з початковою моделлю (початкова інформація про атрибути виробу, обладнання, технології тощо), що є вхідною інформацією для виробничої системи проектування та виготовлення і бере участь у прийнятті рішення на перших етапах проектування виробу.

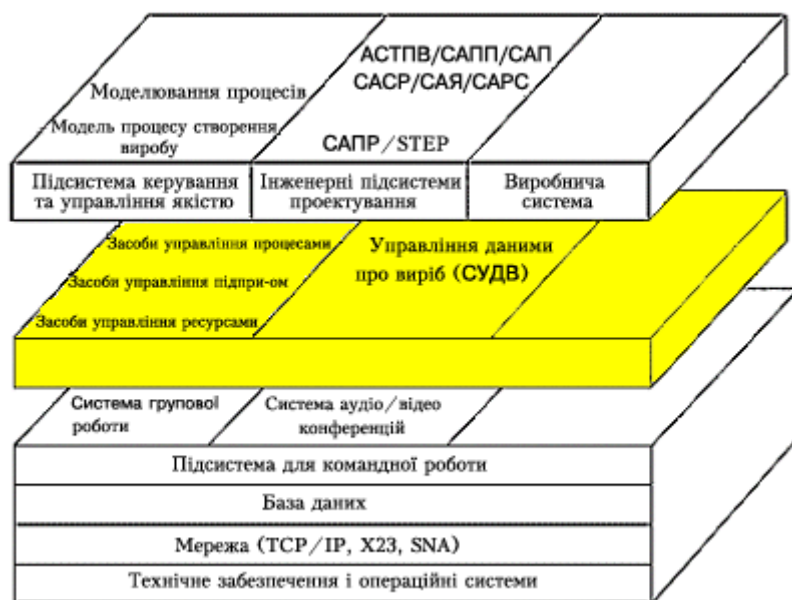


Рис. 6. Архітектура підсистем ІКВ при методології паралельного проектування

Приклад наскрізної інтегрованої інженерної підсистеми проектування на всьому ЖЦВ у ПП показаний на рис. 7. Мета цієї підсистеми – підтримка інформаційної та функціональної інтеграції під час проектування виробів і проектування виробництва. Ця підсистема включає систему управління даними про виріб для підтримки обміну інформацією між усіма учасниками колективу розробників. Властивості наскрізної інтегрованої інженерної підсистеми проектування є такими:

Використання стандарту обміну даними про виріб у форматі STEP для побудови цілісної інформаційної моделі виробу і її збереження як документів, доступних для обміну або безпосереднього зберігання в базі даних. Повідомлення про проектні зміни на будь-якому кроці проектування можуть бути спроекційовані на інші деталі через інтегровану схему, і, при цьому, забезпечується відповідність і несуперечність даних;

Лекція 6. Маршрут автоматизованого проектування

На етапі проектування компонентів виробу визначається можливість складання виробу, його відтворюваність та оптимізація параметрів, використовуючи лише засоби DFX, системи моделювання та інших програмних засобів. Це допомагає уникнути перепроєктування на останньому етапі створення виробу;

Процеси проектування зверху-вниз, які використовуються у системах класу САП, САСР, і АСТПВ, можуть повністю використовувати біжучу інформацію про виріб для виконання проектних робіт. Їхні зворотні потоки надходять до проєктувальників, які працюють із системами вищого рівня (у плані етапів виготовлення виробів), для оцінювання і модифікації. Це може суттєво скоротити цикли виготовлення виробу.

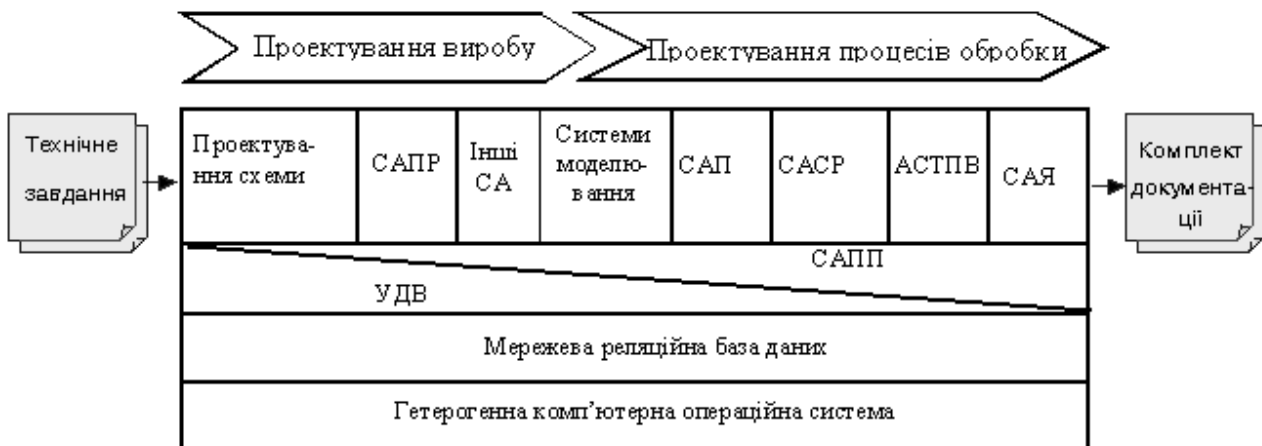


Рис. 7. Приклад наскрізної інтегрованої інженерної підсистеми проектування на всьому ЖЦВ у ПП.

8. Системні середовища і програмно-методичні комплекси

Вступ

САПР відносяться до числа найбільш складних і наукомістких автоматизованих систем (АС). Поряд з виконанням власне проектних процедур необхідно автоматизувати також керування проектуванням, оскільки сам процес проектування стає усе більш складним і найчастіше здобуває розподілений характер. На великих і середніх підприємствах помітна тенденція до інтеграції САПР із системами керування підприємством і документообігу. Для керування настільки складними інтегрованими системами в їхньому складі мається спеціальне ПО — системне середовище САПР або АС.

Перші системні середовища САПР, що називалися моніторними підсистемами або Framework (FW), з'явилися на рубежі 70...80-х рр. У даний час основними функціями системних середовищ САПР є керування даними, керування проектуванням, інтеграція ПО, реалізація інтерфейсу з користувачем САПР, допомога в розробці і супроводі ПО САПР.

Термін Framework стосовно до системних середовищ САПР був введений у 1980 р. фірмою Cadence — одним з піонерів у створенні системних середовищ САПР. Крім Cadence, тематикою Frameworks для САПР електронної промисловості займалося трохи ведучих у цій області фірм (Mentor Graphics, IBM, DEC, Sun Microsystems і ін.), що створили міжнародну асоціацію CFI (CAD Framework Initiative). Широку популярність одержали такі системні середовища, як Falcon Framework фірми Mentor Graphics, Design Framework-2 фірми Cadence і JCF (Jessy-Common Framework) європейської програми ESPRIT.

Важливо відзначити, що проблема системних середовищ САПР, що зародилася в процесі становлення САПР електронної промисловості, одержала розвиток при реалізації CALS-технології в різних галузях машинобудування.

Поняття середовища проектування

Реалізація синхронної співпраці в розподілених САПР можлива шляхом розробки спільного робочого середовища проектування. Спільне проектне середовище - це інструментальний засіб, через який повинна здійснюватись комунікація між учасниками проекту. Воно не тільки формує гнучку і ефективну візуальну комунікацію, але і забезпечує посередництво, при якому один конструктор може зрозуміти модель/проект іншого. Таким чином, в основі представлення проектних елементів у спільному проектному просторі лежить їх спільне використання.

Розробляючи теорію спільного проектного середовища, спеціалістами було виділено дві задачі:

- (1) середовище, в якому здійснюється проектування і зберігання спільних даних,
- (2) середовище, в якому САПР використовує/розподіляє пам'ять і організовує взаємодію під-процесів.

Класичне середовище проектування є візуальним, а проектна інформація представляється у вигляді креслень, ескізів, приміток, діаграм, графів, матриць, рівнянь, таблиць тощо.

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

Для сучасних САПР розподіленого проектування виникає нова проблема - багатокористувацького візуального інтерфейсу. Тому, системи проектування повинні мати додаткові графічні компоненти, які б забезпечували можливість організації багатокористувацького інтерфейсу для колективної роботи.

Звідси, створення справжніх веб-систем проектування пов'язане з розвитком розподіленого і групового проектного середовища. Таке середовище повинно не тільки автоматизувати індивідуальні задачі у формі традиційних інструментів машинного моделювання, але й надавати можливість індивідуальним членам проектної групи спілкуватися з один одним, ділитися інформацією, знаннями, співпрацювати і координувати свої дії в межах контексту відповідного проекту.

Основні задачі середовища проектування

Розподілена реалізація проектних процедур відображає специфіку організації проектної групи, а саме - об'єднання територіально розділених членів команди у віртуальну групу на основі єдиної моделі проекту. Для цього, проектувальники повинні інтерактивно працювати через централізований сервер, який реалізує модель проекту.

Основною задачею розподілених групових САПР є їх здатність працювати з, так званим, віртуальним прототипом об'єкту проектування. Мета віртуального прототипування - побудувати повний віртуальний макет в такому вигляді, що проектні і виробничі проблеми стануть очевидними і передбачуваними й можуть бути обговорені й розв'язані в рамках спільного розподіленого робочого середовища. Таке розподілене середовище САПР, де віртуальний прототип розповсюджується через різні мережі з різними платформами, операційними системами як серед команд проектувальників, так і не користувачів САПР, було запропоноване в роботі (рис. 1). В процесі проектування, дані мігрують через різні системи баз даних і мережі надзвичайно динамічно, що виходить за межі простого веб-перегляду. Модель для розподілених об'єктів САПР повинна дозволити різним командам працювати спільно, маючи доступ та змінюючи інформацію в розподілених інженерних системах у реальному часі.

Лекція 8. Системні середовища і програмно-методичні комплекси

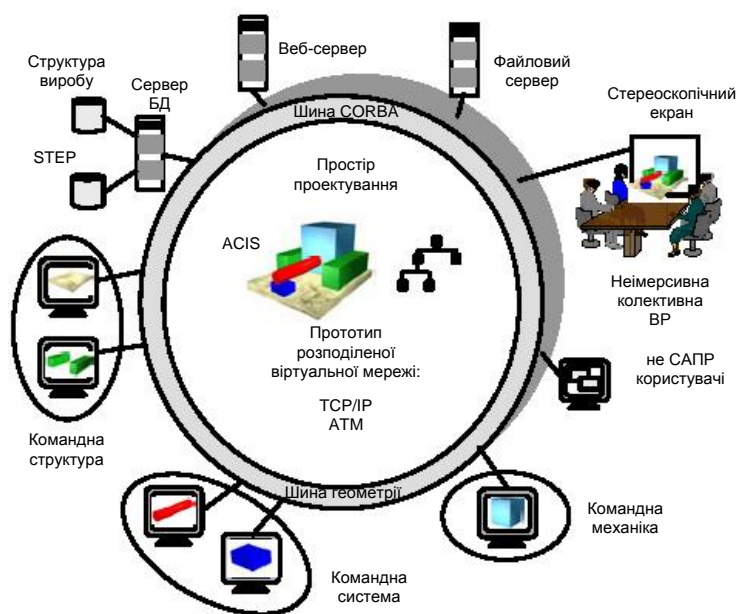


Рис. 1. Розподілене середовище для САПР.

Однак, загальний підхід в розподілених САПР - це аналогія використання розподілених БД. Специфіка технічних засобів і програмного забезпечення полягає в тому, щоб підтримувати комунікацію і багатокористувацький доступ до електронної моделі даних проекту. Електронна модель даних може бути реалізована як у вигляді загальних файлів, так і відкритого доступу до БД. Тому, для розподілених колективних САПР основною умовою є те, що доступ до даних проекту відбувається через наперед розроблене середовище проектування.

Колективне проектування здійснюється групою проектувальників, що працюють однією командою у спільному представленні проектних вимог, креслень і документів. Синхронна співпраця означає, що співпраця відбувається коли всі члени проектної команди працюють одночасно над єдиною документацією / інформацією / проблемою. Цей вид співпраці вимагає розширеної взаємодії між проектувальниками в одному або різних доменах. Розширене поняття колективної співпраці проекту передбачає, що комп'ютерна підтримка такої діяльності повинна забезпечувати гнучкість в обміні проектними даними та ідеями.

Модель середовища проектування

Модель представлення об'єкту проектування в розподілених системах повинна включати декілька компонентів: геометричне представлення у такій формі, щоб система проектування могла його обробляти та візуалізувати на екрані робочих станцій; геометричне представлення, яке дозволяє формувати та обробляти конфліктні форми; комплексне представлення проектного рішення з інформацією про схеми та інтенст.

Структурна схема синхронного колективного проектування, яка запропонована в роботі [19], представлена на рис. 1.6. Модель запропонованого середовища проектування передбачає, що всі існуючі програми можуть бути одночасно відкриті декількома учасниками процесу проектування. Враховуючи вимоги, що висувають до колективного середовища проектування, дана модель представляє дві категорії робочого середовища: спільне візуальне представлення та спільна обробка проектних даних.

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

Користувачі здатні бачити проектні рішення один одного, що забезпечується одночасним візуальним представленням, а також використовувати спільні результати проектування, що забезпечується шляхом спільного представлення проектних даних через сесії централізованого сервера. Чотири компоненти цієї моделі включають:

Сесійний сервер – стартує процес проектування та відповідає за підтримку сесій колективного проекту.

Координатор – спеціальний додатковий процес, який втілює функції управління і контролює обмін даними між додатками і колективним середовищем.

Спільне візуальне представлення – система візуального представлення спільних елементів проекту з можливістю візуалізації конфліктних елементів.

Спільне базове представлення – система, що включає множину моделей об'єктів і процесів, які забезпечують вирішення конфліктних ситуацій, а також забезпечують специфічні проектні рішення при зображенні елементів проекту та їх взаємозв'язків.

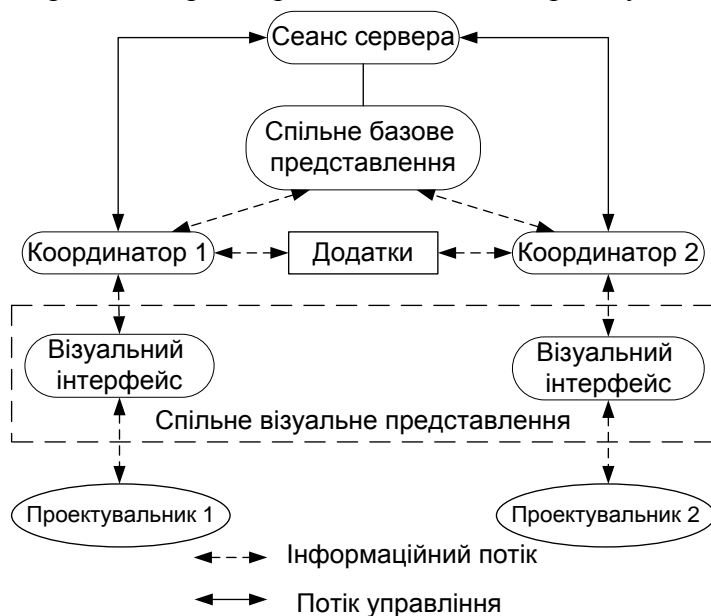


Рис. 2. Модель синхронного колективного середовища проектування.

4. Призначення і склад системних середовищ САПР

У типовій структурі ПО системних середовищ сучасних САПР можна виділити наступні підсистеми.

Лекція 8. Системні середовища і програмно-методичні комплекси



Ядро відповідає за взаємодію компонентів системного середовища, доступ до ресурсів ОС і мережі, можливість роботи в гетерогенному середовищі, налаштування на конкретну САПР (конфігурування) за допомогою спеціальних мов розширення.

Підсистема керування проектом, називана також підсистемою наскрізного рівнобіжного проектування CAPE (Concurrent Art-to-Product Environment), виконує функції спостереження за станом проекту, координації і синхронізації паралельно виконуваних процедур різними виконавцями. Прикладами підсистем керування проектами в машинобудуванні можуть служити Design Manager у САПР Euclid, UG/Manager у Unigraphics. Іноді в окрему підсистему виділяють керування методологією проектування. При цьому під *методологією* розуміють сукупність методів і засобів утворення *маршрутів* проектування — послідовностей проектних операцій і процедур, що ведуть до мети проектування.

Методи побудови маршрутів проектування (workflow) залежать від типу проектних задач. Розрізняють прості задачі, виконувані одною програмою, лінійні, у яких немає розгалужень у межпрограмних зв'язках, і комплексні. Методи побудови маршрутів можуть бути засновані на попередньому описі задач або на попередньому описі правил конструювання задач. В описі задач фігурують порти, з якими співвіднесені дані. Порти можуть бути обов'язковими і необов'язковими, що породжують додаткового даного або дані нового об'єкта. Опису задач даються у виді графів або на мовах розширення.

Підсистема керування методологією проектування представлена у виді бази знань. У цій базі утримуються такі зведення про предметну область, як інформаційна модель (наприклад, у виді діаграм сутність-відношення), ієрархічна структура проєктованих об'єктів (наприклад, у виді І-або-дерева), опису типових проектних процедур, типові фрагменти маршрутів проектування — так називані потоки процедур, відповідність між процедурами і наявними пакетами прикладних програм, обмеження на їхнє застосування і т.п. Часто таку БЗ доповнюють навчальною підсистемою, використовуваною для підготовки фахівців до використання САПР.

Сучасні **системи керування проектними даними** називають PDM (Product Data Manager), іноді стосовно до АСУ використовують назва EDM (Enterprise Data Manager). PDM призначені для інформаційного забезпечення проектування і виконують наступні функції:

- збереження проектних даних і доступ до них, у тому числі ведення розподілених архівів документів, їхній пошук, редагування, маршрутизація і візуалізація;

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

- керування конфігурацією виробу, тобто ведення версій проекту, керування внесенням змін;
- створення специфікацій;
- захист інформації;
- інтеграція даних (підтримка типових форматів, конвертування даних).

Основний компонент PDM — *банк даних* (БНД). Він складається із системи керування базами даних і баз даних (БД). Міжпрограмний інтерфейс значною мірою реалізується через інформаційний обмін за допомогою банку даних. PDM відрізняє легкість доступу до ієрархічно організованим даних, обслуговування запитів, видача відповідей не тільки в текстової, але й у графічній формі, прив'язаної до конструкції виробу. Оскільки взаємодія усередині групи проектувальників в основному здійснюється через обмін даними, то в системі PDM часто сполучають функції керування даними і керування рівнобіжним проектуванням.

Підсистема інтеграції ПЗ призначена для організації взаємодії програм у маршрутах проектування. Вона складається з ядра, що відповідає за інтерфейс на рівні підсистем, і оболонок процедур, що погодять конкретні програмні модулі, програми і/або програмно-методичні комплекси (ПМК) із середовищем проектування.

Інтеграція ПЗ базується на ідеях об'єктно-орієнтованого програмування. Варто розрізнити синтаксичний і семантичний аспекти інтеграції. Синтаксична інтеграція реалізується за допомогою уніфікованих мов і форматів даних, технологій типу ODBC для доступу до загального банку даних або компонентно-орієнтованих (CBD — Component-Based Development) технологій. Приклад уніфікованого формату — TES (Tool Encapsulation Specification), запропонованого консорціумом CFI. Інформація з TES використовується для створення оболонок модулів при інкапсуляції. Семантична інтеграція має на увазі автоматичне розпізнавання різними системами змісту переданих між ними даних і досягається значно сутужніше.

Підсистема користувальницького інтерфейсу містить у собі текстовий і графічний редактори і підтримується системами багатівіконного інтерфейсу типу X Window System або Open Look.

Підсистема CASE призначена для адаптації САПР до нестатків конкретних користувачів, розробки і супроводи прикладного ПО. Її можна розглядати як спеціалізовану САПР, у якій об'єктом проектування є нові версії підсистем САПР, зокрема, версії, адаптовані до вимог конкретного замовника. Іншими словами, такі CASE-підсистеми дозволяють користувачам формувати порівняно з малими витратами зусиль варіанти прикладних ПМК із наявного базового набору модулів під заданий вузький діапазон конкретних умов проектування. У таких випадках CASE-підсистеми називають *інструментальними середовищами*.

CASE-система, як система проектування ПО, містить компоненти для розробки

Лекція 8. Системні середовища і програмно-методичні комплекси

структурних схем алгоритмів і "екранів" для взаємодії з користувачем в інтерактивних процедурах, засобу для інфологічного проектування БД, налагодження програм, документування, збереження "історії" проектування і т.п. Поряд з цим, у CASE-підсистему САПР входять і компоненти зі специфічними для САПР функціями.

Так, до складу САПР Microstation (фірма Bentley Systems) включене інструментальне середовище Microstation Basic і мова MDL (Microstation Development Language) з відповідною програмною підтримкою. Мова MDL — 3-подібний, з його допомогою можна лаконічно виразити звертання до проектним операціям і процедурам. У цілому середовище Microstation Basic близьке за своїми функціями до середовища MS Visual Basic, у ній мають генератор форм, редактор, конструктор діалогу, відлагоджувач.

САПР Спрут (російська фірма Sprat Technologies) узагалі створена як інструментальне середовище для розробки користувачем потоків задач конструкторського і технологічного проектування в машинобудуванні з наступним можливим оформленням потоків у виді користувальницьких версій САПР. Сконструйований потік підтримується компонентами системи, у число яких входять графічні 2D і 3D підсистеми, СУБД, продукційна експертна система, документатор, технологічний процесор створення програм для верстатів із ЧПУ, постпроцесори.

Найбільш відомою CASE-системою в складі САПР у даний час є описувана нижче система CAS.CADE фірми MatraDatavision, з її допомогою фірма розробила чергову версію Euclid Quantum своєї САПР Euclid.

Підходи до інтеграції ПЗ в САПР.

Для створення ПЗ САПР так само, як і інших складних автоматизованих інформаційних систем, що визначає значення мають питання інтеграції ПЗ. Теоретичною базою для створення технологій інтеграції ПЗ в САПР є:

методологія автоматизованого проектування, відповідно до якої здійснюються типізація проектних процедур і маршрутів проектування в різних предметних областях, виявлення типових вхідних і вихідних даних процедур, побудова інформаційних моделей додатків і їхнє узагальнення, порівняльний аналіз альтернативних методів і алгоритмів виконання типових процедур;

об'єктно-орієнтованна методологія, відповідно до якої множині сутностей, фігуруючих у процесах проектування, підрозділяються на класи, у класах з'являються свої процедури і типи даних з відносинами спадкування. Ці класи можуть бути інваріантними і прикладними. Їхнє узагальнення й уніфікація приводять до появи таких понять і засобів, як інтегровані ресурси і прикладні протоколи, що фігурують у стандартах STEP, або уніфіковані програмні компоненти типу графічних ядер конструкторських САПР. Саме наявність типових процедур і однакове тлумачення атрибутів об'єктів у рамках конкретних протоколів дозволяють різним програмним системам "розуміти" один одного при взаємодії.

Поряд з типовими графічними ядрами, відомі типові ПМК імітаційного моделювання, конструювання деталей і механізмів, технологічної підготовки виробництва й ін. Можливість використання типових програм у складі програмних комплексів обумовлена саме уніфікацією інтерфейсів при обмінах даними.

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

У деяких маршрутах проектування обміни даними повинні відбуватися з високою частотою, що обумовлює специфічні вимоги до інтерфейсів. Прикладом можуть служити задачі імітаційного моделювання, у яких потрібно імітувати взаємодія процесів, описуваних за допомогою різного МО (наприклад, на зосереджених і розподіленому ієрархічних рівнях, або за допомогою аналогових і дискретних моделей). Для таких задач при моделюванні характерне відтворення часової послідовності подій, що відбуваються в аналізованих взаємодіючих системах. Відповідно взаємодія програм моделювання може відбуватися через фіксоване число часових кроків або в міру здійснення тих або інших подій у моделюючих системах.

Так, у програмах змішаного аналого-дискретного моделювання електронних пристроїв аналогова частина моделюється за допомогою програми аналізу електронних схем, а дискретна частина — за допомогою програми логічного моделювання. Вплив аналогової частини на дискретну відображається в математичних моделях шляхом перетворення безперервних фазових перемінних у логічні перемінні в місцях сполучення частин моделі, зворотний вплив виражається в перетворенні ідеалізованих логічних сигналів у задані функції часу, що відповідають електричним сигналам заданої форми. Очевидно, що в змістовній частині повідомлень, переданих з однієї частини в іншу, повинні бути зведення або про стани, що виражаються значеннями фазових перемінних у інтерфейсних вузлах, або про події — змінах фазових перемінних. Обмін повідомленнями може відбуватися багаторазово протягом акта одноваріантного аналізу.

У програмно-методичних комплексах конструювання відбувається обробка графічної інформації. Змістова частина повідомлень відноситься до геометричних елементів, їхнім розмірам і положенню в просторі. У програмах технологічної підготовки механічної обробки деталей поряд з геометричною інформацією про конструкції заготовель у передані повідомлення можуть входити зведення про інструмент, технологічне оснащення, устаткування, режимах обробки, нормах часу, траєкторіях руху інструмента і робочих органів устаткування і т.п.

Іншими словами, у кожному додатку сукупність використовуваних при обмінах понять, предметних перемінних і числових параметрів істотно обмежена і достатно визначена для того, щоб можна було порушувати питання про типізацію моделей і мови взаємодії. Такі питання вирішуються в рамках технологій STEP/CALS. Число додатків, що знайшли свій опис у прикладних протоколах STEP обмежено, але сукупність таких протоколів може розширюватися.

Прикладні протоколи STEP представляють семантичну сторону інтеграційних технологій. Для інтеграції потрібна не тільки уніфікація моделей додатків, але й уніфікація механізмів взаємодії, прикладами яких є технології OLE, DDE, а також компонентно-орієнтовані технології.

Технології інтеграції ПЗ типу DDE і OLE.

Сучасні ОС дозволяють працювати одночасно з декількома задачами з виділенням кожній задачі свого вікна на екрані дисплея. Міжпрограмні взаємодії здійснюються шляхом

Лекція 8. Системні середовища і програмно-методичні комплекси

посилки повідомлень, як це прийнято в об'єктно-орієнтованому програмуванні. Використовуються спеціальні засоби організації взаємодій.

Так, ОС Unix підтримує взаємодія асинхронних рівнобіжних процесів, у тому числі в різних вузлах мережі. Кожен клієнт повинний попередньо зафіксувати свої потреби у виді імен використовуваних повідомлень. Повідомлення мають структуру фрейму. Одержувач повідомлення визначає, що повідомлення відноситься до нього, викликає оброблювач повідомлення і використовує отримані дані відповідно до своїх функцій.

В операційних системах Microsoft для організації міжпрограмних взаємодій були запропоновані засоби Clipboard, DDE, OLE і надалі технологія Active..

Робота Clipboard заснована на традиційному способі обмінних зон — виділенні кишені (деякої області оперативної пам'яті, поділюваної взаємодіючими програмами). При обмінах одна програма посилає повідомлення в кишеню, а інша витягає, інтерпретує і використовує це повідомлення. Аналогічний режим роботи здійснюється за допомогою технології формування складених документів OLE, але тут розширені можливості комбінування даних різних типів у переданих документах.

Розрізняють два способи взаємодії: зв'язок (linking) і впровадження (embedding). При зв'язку в створюваний документ включається не сам текст із джерела, а лише посилання на нього. Очевидно, що тут менше витрати пам'яті, зміни в джерелі автоматично переходять у документ. При впровадженні текст із джерела фізично переноситься в документ. Після чого документ можна редагувати незалежно від джерела. Обоє цих способу реалізовані в технології OLE, що і зафіксовано в її назві (OLE — Object Linking and Embedding).

При обміні за допомогою DDE (Dynamic Data Exchange) програма-клієнт запитує режим діалогу з програмою-сервером. У повідомленні вказується ім'я сервера, ім'я роздягнула (звичайно роздягнув — це файл), ім'я елемента (обмінювана порція інформації). Попередньо такий елемент (атом) повинний бути створена, його адреса зафіксована в таблиці атомів. У відповідь на запит створюється канал, по якому сервер передає дані або, що реалізується частіше, пересилає адресу потрібного атома. По цій адресі клієнт додатковою командою може одержати дані.

Підхід до реалізації інтероперабельності, наявний у DDE і OLE, одержав розвиток у сучасних компонентно-орієнтованих технологіях розробки ПО, розглянутих нижче.

Керування даними в САПР.

У більшості автоматизованих інформаційних систем застосовують СУБД, що підтримують реляційні моделі даних.

Серед загальних вимог до СУБД можна відзначити: 1) забезпечення цілісності даних (їхньої повноти і вірогідності); 2) захист даних від несанкціонованого доступу і від перекручувань через збої апаратури; 3) зручність користувальницького інтерфейсу; 4) у більшості випадків важлива можливість розподіленої обробки в мережах ЕОМ.

Перші дві вимоги забезпечуються обмеженням прав доступу, заборонаю одночасного використання тих самих оброблюваних даних (при можливості їхньої модифікації), введенням контрольних точок (checkpoints) для захисту від збоїв і т.п.

Банк даних у САПР є важливою обслуговуючою підсистемою, він виконує функції

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

інформаційного забезпечення і має ряд особливостей. У ньому зберігаються як рідко змінювані дані (архіви, довідкові дані, типові проектні рішення), так і зведення про поточний стан різних версій виконуваних проектів. Як правило, БНД працює в багатокористувацькому режимі, а з його допомогою здійснюється інформаційний інтерфейс (взаємодія) різних підсистем САПР. Побудова БНД САПР — складна задача, що обумовлено наступними особливостями САПР:

Розмаїтість проектних даних, що фігурують у процесах обміну як по своїй семантиці, так і по формах представлення. Зокрема, значна частка графічних даних.

Нерідко обміни повинні вироблятися з високою частотою, що пред'являє тверді вимоги до швидкодії засобів обміну (думають, що СУБД повинна працювати зі швидкістю обробки тисяч сутностей у секунду).

У САПР проблема цілісності даних виявляється більш важкої для рішення, чим у більшості інших систем, оскільки проектування є процесом взаємодії багатьох проектувальників, що не тільки зчитують дані, але і змінюють них, причому значною мірою працюють паралельно. З цього факту випливають наслідки: по-перше, ітераційний характер проектування звичайно приводить до наявності по кожній частині проекту декількох версій, кожна з них може бути прийнята надалі в якості основний, тому потрібно зберігати усі версії з можливістю повернення до кожної з них; по-друге, не можна допускати використання незатверджених даних, тому проектувальники повинні мати свій робочий простір у пам'яті і працювати в ньому автономно, а моменти внесення змін у загальну БД повинні бути погодженими і не породжувати для інших користувачів невизначеності даних.

Транзакції можуть бути тривалими і трудомісткими. *Транзакцією* називають послідовність операцій по задоволенню запиту. У САПР внесення змін у деяку частину проекту може викликати досить довгу і розгалужену мережу змін в інших його частинах через суттєву взаємозалежності компонентів проекту (багатокроковість реалізації запитів). У частковому випадку, транзакції можуть містити в собі такі трудомісткі операції, як верифікація проектного рішення за допомогою математичного моделювання. У результаті транзакції можуть тривати навіть декілька годин і більше. Одні з труднощів полягає у відображенні взаємозалежності (асоціативності) даних. При збереженні компонентів проекту в зовнішній пам'яті витрати часу на опрацювання запитів виявляються значно вище, ніж у більшості інших автоматизованих систем, з менш вираженими взаємозалежностями даних.

Ієрархічна структура проектних даних і, отже, відображення спадкування в цілях скорочення обсягу бази даних.

У визначеній мері названі особливості враховуються в СУБД третього покоління, у яких стали застосовуватися риси об'єктно-орієнтованих (об'єктних) СУБД. У них набори даних, що характеризують стан предметної області (стан проекту у випадку САПР), містяться в окремі файли. Інтерпретація семантики даних здійснюється за допомогою спеціальних процедур (методів), що супроводжують набори. Спадкування властивостей об'єктів предметної області виражається за допомогою введення категорій класу, надкласу, підкласу. Інформаційні моделі додатків для таких СУБД розробляються на основі методик типу IDEF1X.

Об'єктні БД вигідні, по-перше, тим, що дані по конкретних об'єктах проектування не

Лекція 8. Системні середовища і програмно-методичні комплекси

розкидані по множині таблиць, як це має місце в реляційних БД, а зосереджені у визначених місцях. По-друге, для кожного об'єкта можуть бути призначені свої типи даних. У результаті простіше вирішуються задачі керування і задоволення запитів.

Поряд з чисто об'єктними СУБД (pure ODBMS), застосовують СУБД об'єктно-реляційні. В останніх відбувається об'єднання властивостей реляційних і об'єктно-орієнтованих СУБД: об'єктно-орієнтована СУБД забезпечується непроцедурною мовою запитів або в реляційну СУБД вводяться спадкування властивостей і класи. Непроцедурність вхідної мови забезпечується використанням мови SQL. Її оператори безпосередньо включаються в програми мовою С. Можливе написання додаткових програм, що інтерпретують SQL-запити.

Відмінні риси СУБД третього покоління: розширений набір можливих типів даних (це абстрактні типи, масиви, множини, записи, композиції різних типів, відображення величин зі значеннями різних типів), відкритість (приступність з різних мов програмування, можливість звертання до прикладних систем із СУБД), непроцедурність мови (загальноприйнятим стає мова запитів SQL), керування асинхронними рівнобіжними процесами, стан яких відбиває БД. Остання властивість дозволяє говорити про тісний взаємозв'язок СУБД і підсистеми керування проектами DesPM.

Названі особливості керування даними в САПР знайшли своє вираження в сучасних підсистемах керування проектними даними PDM.

У PDM розмаїтість типів проектних даних підтримується їхньою класифікацією і відповідним виділенням груп з характерними множинами атрибутів. Такими групами даних є описи виробів з різних точок зору (аспекти). Для більшості САПР машинобудувань характерними аспектами є властивості компонентів і зборок (ці зведення називають Bill of materials — BOM), моделі і їхнє документальне вираження (основними прикладами можуть служити креслення, 3D моделі візуалізації, сіткові представлення для кінцевого-елементного аналізу, текстові описи), структура виробів, що відбиває взаємозв'язок між компонентами і зборками і їх описами в різних групах.

Унаслідок великого обсягу проектних даних і наявності ряду версій проектів PDM повинна мати розвинуту систему пошуку потрібних даних за різними критеріями.

Розглянуті особливості банків даних у САПР дозволяють кваліфікувати їх як системи Data Warehouse (DW), тобто сховища даних. Для сховищ даних характерний ряд особливостей, що збігаються з названими вище особливостями банків даних САПР: 1) тривале збереження інформації, що відбиває історію розробок; 2) частота операцій читання даних вище частоти операцій відновлення даних; 3) використання єдиних форматів для однотипних даних, отриманих з різних джерел (наприклад, від різних програмно-методичних комплексів). Ці особливості дозволяють керувати конфігурацією проектів, що, зокрема, означає збереження в САПР усіх версій проекту і, можливо, даних по проектах попередніх розробок, задоволення складних запитів, для відповіді на які потрібне витяг і обробка даних з різних частин сховища (так називана багатомірна обробка). Моделі даних у DW відрізняються від реляційних моделей (RM): у RM використанням нормальних форм прагнуть максимально зменшити надмірність даних, що приводить до збільшення числа таблиць, але зменшених розмірів, однак багатомірний пошук, що вимагається в DW, у множині таблиць утруднений. Тому в DW частіше використовується модель даних "зірка", у якій мається загальна таблиця фактів (Fact Table) і кожному фактові ставиться у відповідність кілька таблиць з необхідними атрибутами.

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

Цілісність даних у DW забезпечується перевіркою і трансформацією даних (data cleaning), що вводяться з зовнішніх джерел, наявністю дисципліни відновлення даних, централізованою збереженням основної бази, при цьому достатня швидкодія підтримується передачею копій визначених частин бази в локальні бази, називані кіосками даних (Data Mart) і орієнтовані на окремі групи користувачів.

Прикладом СУБД, що враховує вимоги, пропоновані з боку САПР, є система IMAN фірми EDS Unigraphics. Це система керування об'єктно-орієнтованими базами даних, її можна також назвати системою інтеграції даних. Вона виконує функції підсистеми PDM, що є функціями збереження даних, керування доступом до них, контролю внесених змін, створення специфікацій виробів, інтегрування прикладних підсистем. Усередині IMAN використовується реляційна модель даних, а на інтерфейсному рівні — об'єктно-орієнтована інформаційна модель. Для синхронізації змін передбачається блокування доступу користувачів, якщо з БД уже початків роботу деякий користувач.

Іншими відомими прикладами підсистем керування проектними даними можуть служити системи Optegra (фірма Computervision), Euclid Design Manager (Matra Datavision), ProPDM у складі САПР Pro/Engineer (PTC), TechnODOCS (Російська фірма "Звістка").

Ряд фірм розробляє системи PDM, які можна використовувати як самостійні продукти і як підсистеми в автоматизованих системах проектування і керування.

Прикладом може служити система Part (фірма Люція Софт), у якій передбачені функції керування конфігурацією виробів, керування проектними даними і документообігом, графічний користувальницький інтерфейс, реалізація архітектури сервер-сервер-клієнт-сервер.

Варіанти керування даними в мережах АС

При мережній організації АС інформаційне забезпечення може бути реалізоване по одному з наступних варіантів: 1) FS — файловий сервер; 2) RDA — доступ до вилучених даних; 3) DBS — сервер баз даних; 4) AS — сервер додатків. Варіанти розрізняються розподілом між різними вузлами мережі функцій збереження даних, керування даними, обробки даних у додатках і інтерфейсу з користувачем.

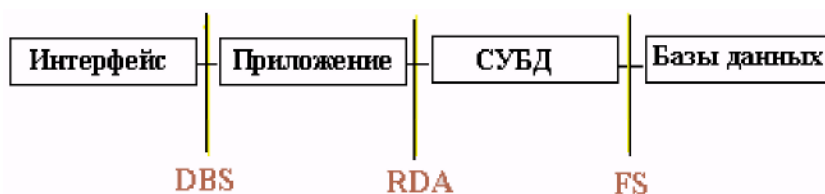


Рис 5.3. Місце середовища передачі даних.

Варіанти двохзіркових схем розподілених обчислень показано вертикальною рисою для перших трьох варіантів. Кожен варіант має свою область застосування.

Варіант файлу-сервера характерний для локальних мереж на персональних ЕОМ з невеликим числом користувачів.

Лекція 8. Системні середовища і програмно-методичні комплекси

У РСУБД із реплікацією немає проблеми узгодження при записі дій багатьох вузлів. Власне тиражування найчастіше виконується за правилом повної еквівалентності — оновлені дані відразу ж після їх зміни розсилаються по всіх локальних БД. Читання ж виконується з БД одного конкретного вузла, найбільш близького до користувача у функціональному або географічному змісті.

Складніше вирішувати проблеми розподіленого керування, що потрібно в РСУБД без тиражування. Одним з розповсюджених протоколів розподіленого керування є протокол двохфазної фіксації транзакцій (2PC). На першій фазі ініціатор транзакції(координатор) розсилає учасникам виконання транзакції оповіщення про блокування. У відповідь вузли повідомляють про свою готовність або неготовність. На другій фазі координатор повідомляє або про "глобальну фіксацію", тобто про виконання транзакції, або про відкит транзакції. Неприємності можливі при збоях, що можуть залишити деякий вузол у заблокованому стані: він не може ні виконувати транзакцію, ні скасовувати неї в однобічному порядку.

Програмні засоби керування проектуванням у САПР.

У залежності від ступеня автоматизації керуючих функцій можна виділити кілька рівнів керування проектуванням:

компонентний; на цьому рівні користувач повинний знати специфічні особливості кожної конкретної програми, використовуваної в маршруті проектування; при організації маршруту він повинний подбати про інформаційні інтерфейси використовуваних програм; іншими словами, системне середовище лише надає зведення про наявні програми і їхні інтерфейси;

ресурсний; користувач як і раніше оперує програмами при компіляції маршруту проектування, але системне середовище дозволяє сховати специфіку кожної програми, тому що спілкування уніфіковано;

задачний; користувач складає маршрут проектування не з окремих програм, а з окремих проектних процедур; покриття маршруту програмами виконує системне середовище;

проблемний; користувач формулює завдання у формі "що потрібно зробити", а не "як це зробити", тобто не визначає маршрут проектування, а ставить проектну проблему.

У системних середовищах САПР *керування проектуванням* покладається на підсистему CAPE, у деяких системах що позначається як DesPM (Design Process Manager). DesPM повинна містити в собі компоненти: комплекси базових знань по тим предметним областям, що визначаються об'єктом проектування, а також знань про мови представлення характеристик і обмежень; засобу для генерації плану (маршруту проектування), визначення наявності засобів і ресурсів для реалізації плану; засобу виконання плану; засобу оцінки результатів. DesPM дозволяє вибирати об'єкти проектування, робити декомпозицію моделей, для кожного компонента вибирати проектні процедури з наявного набору.

По кожнім об'єкті DesPM видає повідомлення, прикладами яких можуть бути: "об'єкт проектується іншим розроблювачем", "проектування передчасне, не виконані попередні процедури", "не підготовлені вихідні дані". Однієї з найважливіших функцій DesPM є допомога в реалізації рівнобіжного проектування. Бажано в DesPM передбачити можливості створення "суперпроцедур" — командних файлів для виконання часто повторюваних

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

фрагментів маршрутів проектування.

Розширення можливостей керування проектуванням і адаптація системного середовища до конкретних САПР зв'язане з застосуванням мов розширення. *Мова розширення* — це мова програмування, що дозволяє адаптувати і набувати системне середовище САПР на виконання нових проектів. Мова розширення повинна забезпечувати доступ до різних компонентів системного середовища, поєднувати можливості базової мови програмування і командної мови, включати засобу процедурного програмування. Для більшості мов розширення базовими є ЛІСП або С.

Так, мова Skill з Design Framework-2 фірми Cadence або мова CCL (CASE Comment Language) фірми Matra Datavision є ЛІСП-подібними, а мова AMPLE з Falcon Framework фірми Mentor Graphics базується на мовах С и ПАСКАЛЬ.

Керування процесом проектування містить у собі велике число дій і умов, що підтримують рівнобіжну роботу багатьох користувачів над загальним проектом. Керування виконується на основі моделей обчислювальних процесів. Використовуються специфікації моделей, прийняті в CASE-системах, наприклад, діаграми потоків даних, орієнтовані графи. Спочатку моделі складають для задачного рівня, а потім система здійснює їхнє покриття. Застосовують також опису на мовах розширення або 4GL. У системному середовищі Ulyses специфікації дані у виді набору модулів із вказівкою умов їхньої активізації, що близько до представлення моделей у системах, керованих знаннями. Так, кожен програмний модуль, що проектує, може бути активізований тільки в тому випадку, якщо вхідні дані готові. Для цього спеціальна *програма керування модулями* системного середовища відслідковує дотримання відносин проходження між проектними операціями і процедурами, заданими в маршруті проектування. На цю же програму покладаються функції регулювання прав доступу до модулів, збір статистики (протоколювання) по звертаннях до модулів і деякі інші.

Необхідне забезпечення синхронізації зміни даних, поділюваних багатьма користувачами. Для цього, по-перше, користувачі підрозділяються на класи (адміністрація системи, керівництво проектом і частинами проекту, групи виконавців-проектувальників) і для кожного класу вводять певні обмеження, зв'язані з доступом до поділюваних даних; по-друге, забезпечують засобу ведення багатьох версій проекту; по-третє, для виконання робіт в окремих галузях рівнобіжного процесу користувачам виділяють свої робочі області пам'яті. Даним можуть привласнюватися різні значення статусу, наприклад, "правильно", "необхідне переобчислення", "затверджено як остаточне рішення" і т.п. Власне синхронізація виконується за допомогою механізмів типу рандеву або семафорів, розглянутих у допомога з рівнобіжних обчислень.

Прикладом підсистеми керування проектуванням у САПР СБІС може служити Minerva, розроблена фахівцями університету Карнегі-Меллона (США). У ній реалізується спадне проектування на основі моделі у виді І-або-дерева. Дерево може бути не цілком визначене до початку проектування і його окремих кущів допрацьовуються в процесі проектування. На кожній ярусі дерева відбувається вибір альтернатив, формування ТЗ для наступного ієрархічного рівня, можливі повернення. У засобах користувальницького інтерфейсу передбачено висвітлювання на екрані фрагментів дерева, по кожній галузі дерева

Лекція 8. Системні середовища і програмно-методичні комплекси

повідомляється про її готовність до пророблення, чи займається нею хтось іншої з розроблювачів і т.п.

У загальному випадку повна формалізація керування проектуванням не може бути досягнута, тому корисну роль грають *системи підтримки рішень*, прийнятих людьми, DSS (Decision Support Systems). Як такі системи часто використовують сховища даних і OLAP-засобу (On-Line Analytical Processing).

Використання сховищ даних має ряд переваг у керуванні великими обсягами даних: мається єдине ядро, що виключає надмірно розгалужені і тривалі транзакції, легше синхронізувати внесення змін, підтримувати єдність форматів даних, зберігати попередні версії і т.п.

OLAP-засоби повинні забезпечувати оперативний доступ до даних, на основі якого виявляються залежності між параметрами (вимірами в багатомірній моделі додатка). У OLAP-системах на реляційних СУБД аналітична обробка, або, іншими словами, багатомірний динамічний аналіз даних вимагає перегляду великого числа записів з різних таблиць. Тому продуктивність виявляється невисокою. У спеціалізованих OLAP-системах, що забезпечують більш швидкий багатомірний аналіз, але з більш істотними обмеженнями на обсяг БД, дані зберігаються у виді гіперкубів або полікубів — багатомірних таблиць з постійним або перемінним числом осередків відповідно. Приклад OLAP системи — Oracle Express, що допомагає менеджерам і аналітикам одержувати дані у виді розрізів таких багатомірних таблиць, готувати звіти, обґрунтовувати рішення.

У складі підсистем керування методологією проектування корисно мати засобу консультування по прийняттю проектних рішень. Вони можуть бути представлені у виді множині модулів, поєднаних гіпертекстовою оболонкою. Кожен модуль містить деяка рада на вибір рішення, подоланню протиріч, що виникають у процесі проектування. Тут доречно використання методів і прийомів рішення винахідницьких задач.

Прикладом програми консультування і прогнозування результатів прийнятих рішень може служити програма Clio у згаданій вище підсистемі Minerva.

Приклади підсистем керування даними і проектуванням

У ряді системних середовищ САПР (насамперед САПР у машинобудуванні) у підсистемах PDM поєднуються функції керування даними і проектуванням. Приклад такої PDM — підсистема Design Manager у САПР Euclid Quantum. Функціями цієї PDM є керування потоками проектних даних, версіями проекту, взаємодією розроблювачів, захист інформації, конфігурування й адаптація версій системи для конкретних користувачів.

Підсистема Design Manager у Euclid Quantum складається з частин користувальницької, адміністратора і керування структурою продукту.

У користувальницькій частині дані при виконанні проектування можуть знаходитися або в розпорядженні конкретного розроблювача, зокрема, у його індивідуальної БД (User Area), або в зоні роботи робочої групи (Workgroup Area), зокрема, у її БД. Затверджені дані пересилаються в центральну БД (Repository). Пересилання даних з User Area (UA) у Workgroup Area (WGA) відбувається з ініціативи розроблювача командами *check in* або *share*. Перша з них починає процедуру контролю даних, друга забезпечує поділ даних всіма учасниками робочої

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

групи. Контроль даних виконує уповноважений член групи, результатом є або твердження і, отже, напрямок їх у репозиторій *R*, або нетвердження і відправлення даних у *UA* на доробку. Розроблювач може запитувати дані для початку нового проекту по команді *copy out* або для модифікації існуючого проекту по команді *check out* (рис 5.6).



Рис.5.6. Поток даних у PDM Design Manager (САПР Euclid Quantum)

У БД дані організовані ієрархічно, групуються по іменах проектів або по типах даних. Виклик даних з будь-якої БД (*UA*, *WGA*, *DO*) виконується командою *retrieve*, посилка в БД — командою *store*. При звертанні до БД користувач бачить структуру даних (директорію — імена папок і їхніх частин) і визначений аспект даних виділений в директорії проекту. Такими аспектами можуть бути властивості документа (ім'я, автор, дата, статус і т.п.), список версій проекту, *3D* зображення.

У функції адміністратора системи входять упорядкування даних з їх розподілом по дисках, контроль за правами доступу користувачів, зв'язок із зовнішніми системами (керування імпортом/експортом даних) і ін.

У системному середовищі NELSIS CAD Framework маються частини: 1) DMS (Design Management Services) для підтримки ієрархії даних, керування версіями і потоками задач; 2) DMI (Design Management Interface) з функціями відкриття і закриття баз даних, виклику і пересилання даних, доступу до DMS; 3) FUS (Framework User Services), що включає ряд браузерів для візуалізації інформації.

Базова сутність у NELSIS CAD Framework - об'єкт (осередок). Об'єкт складається з декількох примітивів і/або посилань. Об'єкти поєднуються в модулі. У модулі всі об'єкти мають ті самі імена і тип представлення (*view-type*) і є варіантами опису того самого фізичного об'єкта, тобто це версії або поліпшення попередніх варіантів. Об'єкти можуть знаходитися у відносинах еквівалентності один з одним або ієрархії. Кожен модуль має атрибут, що позначає рівень абстракції. Версії нумеруються і їм привласнюється той або інший статус.

Передбачено наступні статуси: 1) *робітника* - об'єкт знаходиться в роботі, його можна модифікувати, у модулі хоча б один об'єкт повинний мати цей статус; у процесі модифікацій нова версія може заміщати стару або стара версія зберігається, одержуючи, наприклад, статус *Backup*; 2) прийнятий (*actual version*) - саме ця версія служить для обміну між об'єктами, автоматично не стирається, її модифікації здійснюються через робочий статус; 3) архівний (*Backup*), 4) породжуваний (*Derived version*) - статус зарезервований для знову створюваних

Лекція 8. Системні середовища і програмно-методичні комплекси

об'єктів, наприклад, при синтезі проектних рішень. Розроблювач сам змінює статус об'єктів.

Будь-яка зміна повинна відбивати у відносинах об'єкта. NELSIS CAD Framework не змінює існуючі відносини, а створює нові. Наприклад, якщо змінюється об'єкт "топологія", то нова версія не успадковує відношення зі схемою, що була отримана екстракцією зі старої топології. Цілісність даних підтримується тим, що не можна одночасно працювати і змінювати той самий об'єкт різним розроблювачам, тому що кожний з них буде працювати зі своєю робочою версією.

Дані проекту можуть знаходитися в декількох БД розподіленого банку даних. Дані однієї частини проекту доступні іншим частинам, що дозволяє виконувати рівнобіжне проектування.

Для інтеграції програмних компонентів у системне середовище (тобто для узгодження за даними цих компонентів із БД середовища) використовуються звичайні модифікації компонента, якщо відомо його код, або створюється оболонка - модульна абстракція.

У NELSIS CAD Framework мається кілька браузерів для спілкування з користувачем. Для кожного браузера може бути відкрите своє вікно.

Design flow browser - показує взаємозв'язок між проектними процедурами, історію одержання об'єкта, список процедур, що можуть бути виконані над об'єктом, дозволяє задавати маршрути проектування, викликати проектні процедури і задавати їхні параметри,

Hierarchy Browser - показує граф ієрархії і місце об'єкта в ній.

Version Browser - показує усі види (viewtypes), статуси і номери версій обраного об'єкта. Він може показувати відносини еквівалентності, тобто об'єкти, що виражають різні аспекти, наприклад, топологію, схему, результати моделювання фізичного об'єкта.

Equivalence Browser показує відносини еквівалентності для обраного об'єкта.

Schema Browser показує сутності і їхні відносини у виді схеми даних, в окремому вікні показуються запити до БД і відповіді на них.

Компонентно-орієнтовані технології в САПР

Основні ідеї компонентно-орієнтованої (об'єктної) технології зі створенням розширених спеціалізованих бібліотек компонентів реалізовані в системі CAS.CADE (Computer Aided Software/ Computer Aided Design Engineering) фірми Matra Datavision.

Система CAS.CADE складається з декількох частин. Основними частинами є бібліотеки класів і інструментальне середовище для створення програмного забезпечення (ПО) технічних і наукових додатків.

Бібліотеки (Object Libraries) у CAS.CADE являють собою спеціалізовані набори заздалегідь розроблених компонентів мовою C++. Сукупність бібліотек має ієрархічну структуру. Базові компоненти відповідають класам об'єктної методології. Прикладами компонентів є рядки, списки, точки, матриці, лінії, поверхні, дерева, вирішувачі рівнянь, оператори сортування, пошуку на графах і т.п. Класи групуються в пакети (Packages), пакети - у набори (Toolkits), набори - у домени (Resource Domains).

У CAS.CADE виділено кілька бібліотек. По-перше, це бібліотеки 2D і 3D моделювання, що включають компоненти для визначення, створення і маніпулювання геометричними моделями. По-друге, ряд бібліотек призначений для зв'язку з ОС і керування даними, для обміну даними з зовнішніми CAD системами, для створення сіткових моделей і ін. Так, до

Розділ № 1. Методологічні аспекти автоматизованого проектування складних об'єктів і систем

складу бібліотеки обміну даними входять конвертори даних з формату CAS.CADE у Express-файл прикладного протоколу AP214 стандарту STEP і назад. Аналогічні конвертори мають для взаємного перетворення даних з формату CAS.CADE в інші популярні в САПР формати IGES і DXF/SAT

Необхідно відзначити, що основні додатки, на які орієнтована CAS.CADE, — це додатка машинної графіки і геометричного моделювання, тому в системі найбільш розвиті бібліотеки графічних і геометричних компонентів.

Геометричне моделювання і візуалізація в CAS.CADE підтримуються відповідної ПО. У це ПО входять бібліотечні набори "Геометрія", "Топологія", "Візуалізація" і ін. Для тестування і демонстрації компонентів перед їхнім вбудовуванням у проектувану прикладну САПР використовуються спеціальна мова, інтерпретатор і переглядач, що складають підсистему "Тестування".

Набір "Геометрія" включає пакети канонічних геометричних елементів і масивів (множин) цих елементів.

Пакети `gp`, `geom2d` і `geom` включають 2D і 3D геометричні елементи (класи), використовувані як сутності в обчислювальних процедурах, у тому числі в таких операціях, як поворот, відображення, масштабування і т.п. Прикладами елементів можуть служити декартові координати, точки, вектори, лінії, окружності, квадратичні криві, сферичні, тороїдальні і конічні поверхні, криві і поверхні Безь'є, B-сплайнів і ін.

Велике число пакетів розроблене для виконання геометричних побудов і метричних розрахунків. Пакети `gee`, `GC`, `GCE2d` включають алгоритми побудови сутностей з елементів пакетів `gp`, `Geom`, `Geom2d`, наприклад, побудова прямих, дуг окружностей, кривих по заданих параметрах таким, як інцидентні точки, центральні точки і радіуси, рівнобіжні або нормальні прямі і т.п.

Набір "Топологія" визначає структури даних, що описують зв'язки (відносини) між геометричними сутностями - класами попереднього набору "Геометрія". До структур топологічних даних відносяться вершини, ребра, лінії каркасних моделей, ділянки поверхні, оболонки - сукупності зв'язаних через ребра ділянок поверхні, тіла - частини простору, обмежені оболонкою, сукупності тіл, у тому числі прості конструкції виду частин циліндра, конуса, сфери, тора. У наборі мають також засоби: 1) для скруглення гострих кутів і крайок, тобто формування галтелей постійного або перемінного радіуса; 2) для підтримки безперервності при сполученні різних поверхонь; 3) для метричних розрахунків - визначення довжин ребер, площ ділянок поверхні, обсягів тіл, центрів мас і моментів інерції.

У підсистему "Тестування" входять командна мова TCL (Test Command Language), на якому задається програма тестування і перегляду бібліотечних компонентів, інтерпретатор TCL і 2D/3D візуалізатор. У TCL мають звичайні для мов програмування команди, такі як присвоєння значення перемінної, організація циклу, умовний перехід, так і спеціальні команди. Серед останніх виділяють базові, геометричні і топологічні команди. Приклади базових команд: затримка при виконанні програми (наприклад, при презентаціях), звертання до файлу, висновок на екран координат і інших параметрів геометричних об'єктів, створення вікон для різних видів, масштабування зображення, його поворот, установка кольору,

Лекція 8. Системні середовища і програмно-методичні комплекси

виділення на екрані одного заданого об'єкта і т.п. За допомогою геометричних команд виконують створення і модифікацію кривих, поверхонь, геометричні перетворення типу повороту або дзеркального відображення, обчислення координат, кривизн, похідних, перебування точок перетинання ліній і поверхонь. Аналогічні дії роблять стосовно топологічних об'єктів за допомогою топологічних команд.

Інструментальне середовище CAS.CADE включає інтегровану оболонку, підсистему проектування користувальницького інтерфейсу, а також ряд багаторазово використовуваних спеціалізованих програм, таких як 2D і 3D моделлери, підсистема керування даними, прикладні програми аналізу і т.п.

Інтегрована оболонка служить для керування версіями і рівнобіжною роботою багатьох користувачів.

Для проектування користувальницького інтерфейсу в CAS.CADE мають спеціальні мови і програмні засоби. Мова проектування діалогу складається з команд створення інтерфейсу і доступу до компонентів.

Створення інтерфейсу включає створення контейнерів і діалогових елементів. Контейнер являє собою екранне вікно, у якому будуть розміщатися елементи. Елементи забезпечують інформування користувача створюваного додатка про виникаючі події, дають можливість користувачеві задавати значення параметрів, вибирати режим роботи і т.п.

Розрізняють ряд видів контейнерів. Серед них контейнери для повідомлень, що попереджають про помилку, що запитують від користувача відповіді типу "так/ні", завдання розмірів або кольору, вибору файлу і т.п.

Прикладами команд проектування діалогових елементів можуть служити команди визначення позиції елемента у вікні, вибору одного елемента з заданої множини, конструювання текстового рядка або меню, фіксації подій, викликаних вибором мишею позиції або пункту меню, і ін.

У структурі прикладної програми, створюваної в середовищі CAS.CADE, можна виділити діалоговий модуль (модуль користувальницького інтерфейсу GUI - Graphic User Interface), модуль зв'язку з прикладною частиною і власне прикладною частиною, що включає відібрані компоненти і БД, що залежить від додатка

Об'єднання використовуваних у додатку компонентів у прикладну програму здійснюється мовою C++ або спеціальній мові опису інтерфейсів, що нагадує мову IDL.. Отже, реалізуються властиві C++ підтримка спадкування й обмеження доступу (компоненти можуть мати статус захисту від несанкціонованого доступу).

За допомогою CAS.CADE створюють спеціалізовані додатки (насамперед спеціалізовані САПР) з порівняно малими витратами часу і засобів.

9. Технічне забезпечення процесу проектування

Структура ТЗ САПР

Вимоги до ТЗ САПР.

Технічне забезпечення САПР містить у собі різні технічні засоби (hardware), що використовують при автоматизованому проектуванні СОС, а саме ЕОМ, периферійні пристрої, мережне устаткування, а також устаткування деяких допоміжних систем (наприклад, вимірювальних), тобто таких, що підтримують процес проектування.

Використовувані в САПР технічні засоби повинні забезпечувати:

виконання всіх необхідних проектних процедур, для яких мається відповідне ПЗ;
взаємодія між проектувальниками й ЕОМ, підтримку інтерактивного режиму роботи;
взаємодія між членами колективу, що виконують роботу над загальним проектом.

Перша з цих вимог виконується при наявності в САПР обчислювальних машин і систем з достатніми продуктивністю і ємністю пам'яті.

Друга вимога відноситься до користувальницького інтерфейсу і виконується за рахунок включення в САПР зручних засобів введення-виводу даних і насамперед пристроїв обміну графічною інформацією.

Третя вимога обумовлює об'єднання апаратних засобів САПР в *обчислювальну мережу*. У результаті загальна структура ТО САПР являє собою мережа вузлів, зв'язаних між собою середовищем передачі даних (рис. 2.1).



Рис. 2.1. Структура технічного забезпечення САПР

Вузлами цієї мережі є робочі місця проектувальників, які часто називають *автоматизованими робочими місцями* (АРМ) або *робочими станціями* (WS — Workstation). Ними можуть бути також великі ЕОМ (мейнфрейми), окремі периферійні і вимірювальні пристрої. Саме в АРМ повинні бути засоби для взаємодії проектувальника з ЕОМ. Що стосується обчислювальної потужності, то вона може бути розподілена між різними вузлами обчислювальної мережі.

Мережеві САПР.

Класифікація обчислювальних мереж САПР

Обчислювальні мережі САПР класифікуються по ряду ознак. У таблиці 2 представлена ця класифікація.

Класифікація обчислювальних мереж САПР

систем

Ознака	Тип зв'язків	Примітка
Топологія зв'язків	Радіальна (зіркоподібна) Кільцева Радіально-кільцева Розподілена	Звичайні дворівневі САПР, у яких існує центральний обчислювальний комплекс і пару АРМів
Склад засобів передачі даних	Однорідна	Складається з програмно-сумісних ЕОМ
Спосіб передачі даних	З каналами, що не комутуються З комутацією каналів З комутацією повідомлень З комутацією пакетів Зі змішаною комутацією	У сеансах зв'язку утворюються транзитні канали між вузлами мережі, що зв'язуються Поетапна передача повідомлень через центри комутації повідомлень Поетапна передача пакетів визначеної довжини Сполучення комутації каналів повідомлень, пакетів
Спосіб керування	Централізована Децентралізована	Керування потоками даних здійснюється центральним вузлом зв'язку Керування потоками даних розподілено по вузлах мережі
Далекість вузлів	Локальна Дистанційна	Відстані між вузлами обмежені заданою величиною L Відстані перевищують величину L

На рис.3. представлено приклад обчислювальної мережі САПР, у якій нижній рівень утворюють комплекси DMS-2, верхній рівень - ЦВК на основі високопродуктивної ЕОМ типу IBM-370.8 комплекси IDS-3 утворюють разом з відповідної DMS-2 радіальну мережу, вузли DMS-2 зв'язані один з одним розподілену мережу.

Приклади підсистем, що проектують

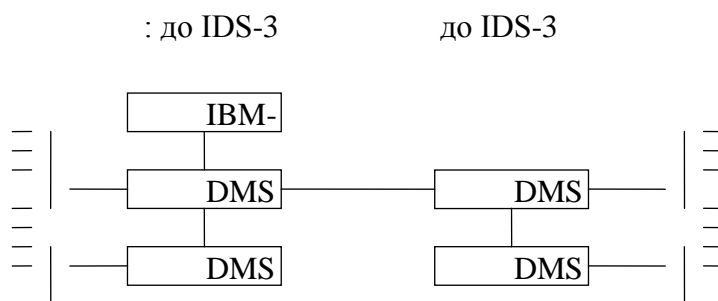


Рис.3. Приклад обчислювальної мережі САПР

У САПР невеликих проектних організацій, що нараховують не більш одиниць-десятків комп'ютерів, що розміщені на малих відстанях один від іншого (наприклад, в одній або

Лекція 9. Технічне забезпечення процесу проектування

декількох сусідніх кімнатах) об'єднуючі комп'ютери мережа є локальною. *Локальна обчислювальна мережа (ЛОМ)* має лінію зв'язку, до якої підключаються усі вузли мережі. При цьому топологія з'єднань вузлів (рис. 2.2) може бути шинна (bus), кільцева (ring) або типу зірка (star). Довжина лінії і число вузлів, що підключаються, у ЛОМ відповідають вимогам до типів мереж в яких функціонують САПР.

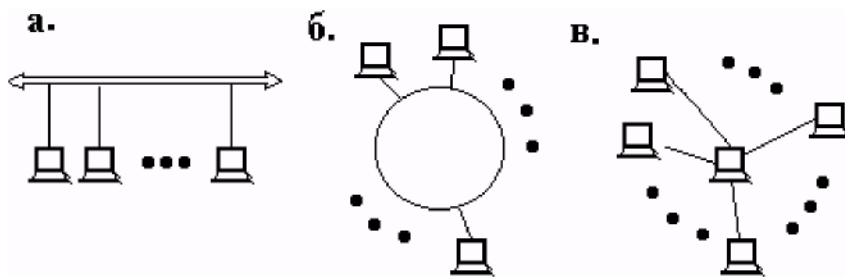


Рис. 2.2. Варіанти

топології локальних обчислювальних мереж: а) шинна; б) кільцева; в) зіркова.

У більш великих по масштабах проектних організаціях у мережу включені десятки-сотні і більш комп'ютерів, що відносяться до різних проектних і управлінських підрозділів і розміщених у приміщеннях одного або декількох будинків. Таку мережу називають *корпоративною*.

Структура ТО САПР для крупній організації представлена на рис. 2.3. Тут показана типова структура великих корпоративних мереж САПР, що називають архітектурою *клієнт-сервер*. У мережах клієнт-сервер виділяється один або декілька вузлів – *серверів*, що виконують у мережі керуючі або загальні для багатьох користувачів проектні функції, а інші вузли (робочі місця) є термінальними. Їх називають *клієнтами*.



Рис. 2.3. Структура

мережевих САПР великих проектних організацій.

Мережі *клієнт-сервер* розрізняють по характеру розподілу функцій між серверами, іншими словами, їх класифікують по типах серверів. Розрізняють *файли-сервери* для збереження файлів, поділюваних багатьма користувачами, *сервери баз даних* автоматизованої системи, *сервери додатків* для рішення конкретних прикладних задач, *комутаційні сервери* (називані також блоками взаємодії мереж або серверами доступу) для взаємозв'язку мереж і підмереж, *спеціалізовані сервери* для виконання визначених телекомунікаційних послуг, наприклад, сервери електронної пошти, тощо.

У випадку спеціалізації серверів по визначених додатках мережу називають *мережею розподілених обчислень*. Якщо сервер додатків обслуговує користувачів однієї ЛОМ, то природно назвати такий сервер локальним. Але оскільки в САПР існують додатки і бази даних, що використовуються користувачами з різних підрозділів і, отже, клієнтами різних ЛОМ, тому відповідні сервери відносять до групи корпоративних, що підключаються зазвичай до опорної мережі (див. рис. 2.3.).

Розподілені обчислювальні мережі САПР

Приклад розподіленої обчислювальної мережі САПР приведений на рис 4.

Розподілена мережа містить локальну обчислювальну мережу кільцевого типу, що поєднує різноманітні ЕОМ і периферійні пристрої:

- - КЭВМ - коллективная ЕОМ,
- - ПК - персональный компьютер,
- - ГД - графічний дисплей,
- - Д - алфавітно-цифровий дисплей,
- - ГП - графобудівник

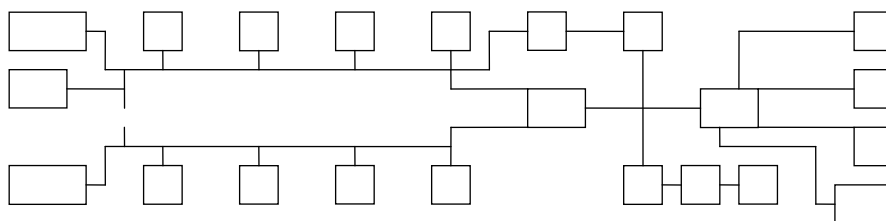


Рис.4. Розподілена обчислювальна мережа САПР:

- МПК - мультиплексор - концентратор, - М - модем, - Т - телефонний канал, - ГАП - ЕОМ гнучкого автоматизованого виробництва, - ВУС - вузол зв'язку.

Кілька входів у мережу об'єднані в мультиплексор-концентратор для передачі даних по виділеному телефонному каналі. На іншій стороні каналу МПК радіально з'єднує ряд різноманітних ЕОМ, у тому числі й ЕОМ гнучкого автоматизованого виробництва.

Один із входів локальної обчислювальної мережі через модем має вихід у телефонну мережу. При цьому забезпечується зв'язок локальної обчислювальної мережі з будь-яким абонентом, що має аналоговий вихід. У цьому випадку можлива координація процесів проектування і виготовлення через канали зв'язку, що комутуються.

Апаратура робочих місць в автоматизованих системах проектування

Обчислювальні системи в САПР.

Як засоби обробки даних у сучасних САПР широко використовують робочі станції, сервери, персональні комп'ютери. Великі ЕОМ і в тому числі суперЕОМ звичайно не застосовують, тому що вони дорогі і їхнє відношення продуктивність/ціна істотно нижче подібного показника серверів і багатьох робочих станцій. На базі робочих станцій або персональних комп'ютерів створюють АРМ.

Типовий склад пристроїв АРМ: ЕОМ з одним або декількома мікропроцесорами, оперативної і кеш-пам'яттю і шинами, що служать для взаємного зв'язку пристроїв; пристрою

Лекція 9. Технічне забезпечення процесу проектування введення-висновку, що включають у себе, як мінімум, клавіатуру, мишу, дисплей; додатково до складу АРМ можуть входити принтер, сканер, плотер (графобудівник), дигитайзер і деякі інші периферійні **пристрої**.

Пам'ять ЕОМ звичайно має ієрархічну структуру. Оскільки в пам'яті великого обсягу важко домогтися одночасно високої швидкості запису і зчитування даних, пам'ять поділяють на надшвидкопрацюючу кеш-пам'ять малої ємності, основну оперативну пам'ять помірного обсягу і порівняно повільну зовнішню пам'ять великої ємності, причому, у свою чергу, кешпам'ять часто розділяють на кеш першого і другого рівнів.

Для зв'язку найбільш швидкодіючих пристроїв (процесора, оперативної і кеш-пам'яті, відеокарти) використовується системна шина з пропускною здатністю до одного-двох Гбайт/с. Крім системної шини на материнській платі комп'ютера маютья шина розширення для підключення мережного контролера і швидких зовнішніх пристроїв (наприклад, шина PCI із пропускною здатністю 133 Мбайт/с) і шина повільних зовнішніх пристроїв, таких як клавіатура, миша, принтер і т.п.

Робочі станції (workstation) у порівнянні з персональними комп'ютерами являють собою обчислювальну систему, спеціалізовану на виконання визначених функцій. Спеціалізація забезпечується як набором програм, так і апаратно за рахунок використання додаткових спеціалізованих процесорів. Так, у САПР для машинобудування переважно застосовують графічні робочі станції для виконання процедур геометричного моделювання і машинної графіки. Ця спрямованість вимагає могутнього процесора, високошвидкісної шини, пам'яті досить великої ємності.

Висока продуктивність процесора необхідна з тієї причини, що графічні операції (наприклад, переміщення зображень, їхні повороти, видалення схованих ліній і ін.) часто виконуються стосовно всіх елементів зображення. Такими елементами в тривимірній (3-D) графіці при апроксимації поверхонь полігональними сітками є багатокутники, їхнє число може перевищувати 10^4 . З іншого боку, для зручності роботи проектувальника в інтерактивному режимі затримка при виконанні команд зазначених вище операцій не повинна перевищувати декількох секунд. Але оскільки кожна така операція стосовно кожного багатокутника реалізується великим числом машинних команд необхідна швидкодія складає десятки мільйонів машинних операцій у секунду. Така швидкодія при прийнятній ціні досягається застосуванням поряд з основним універсальним процесором також додаткових спеціалізованих (графічних) процесорів, у яких визначені графічні операції реалізуються апаратно.

У найбільш могутніх робочих станціях у якості основних звичайно використовують високопродуктивні мікропроцесори зі скороченою системою команд (з RISC-архітектурою), що працюють під керуванням однієї з різновидів операційної системи Unix, Linux. Також використовують мікропроцесори Intel і операційні системи Windows. Графічні процесори виконують такі операції, як, растеризації — представлення зображення в растровій формі для її візуалізації, переміщення, обертання, масштабування, видалення схованих ліній і т.п.

Типові характеристики робочих станцій: кілька процесорів, десятки-сотні мегабайт оперативної і тисячі мегабайт зовнішньої пам'яті, наявність кеш-пам'яті, системна шина із швидкостями від сотень Мбайт/з до 1-2 Гбайт/с.

У залежності від призначення існують АРМ конструктора, АРМ технолога, АРМ керівника проекту і т.п. Вони можуть розрізнятися складом периферійних пристроїв, характеристиками ЕОМ.

В АРМ конструктора (графічних робітників станціях) використовуються растрові монітори з кольоровими трубками. Типові значення характеристик моніторів знаходяться в наступних межах: розмір екрана по діагоналі 17...24 дюйма (фактично зображення займає площа на 5...8 % менше, ніж вказується в паспортних даних).

Відзначимо, що чим нижче частота кадрового розгорнення, а це є частота регенерації зображення, тим помітніше мерехтіння екрана. Бажано, щоб ця частота була не нижче 75 Гц.

Спеціальний випуск ЕОМ як сервери високої продуктивності звичайно мають структуру симетричної багатопроцесорної обчислювальної системи. У них системна пам'ять розділяється всіма процесорами, кожен процесор може мати свою надоперативну пам'ять порівняно невеликої ємності, число процесорів невелике (одиниці, рідко більше десяти).

Периферійні пристрої.

Для введення графічної інформації з наявних документів у САПР використовують дигитайзери і сканери.

Дигитайзер застосовують для ручного введення. Він має вигляд кульмана, по його електронній дошці переміщається курсор, на якому розташований візир і кнопкова панель. Курсор має електромагнітний зв'язок із сіткою провідників в електронній дошці. При натисканні кнопки в деякій позиції курсору відбувається занесення в пам'ять інформації про координати цієї позиції. У такий спосіб може здійснюватися ручна "відколка" креслень.

Для автоматичного введення інформації з наявних текстових або графічних документів використовують *сканери* планшетного або протяжливого типу. Спосіб зчитування — оптичний. У сканувальній головці розміщуються оптоволоконні самофокусовані лінзи і фотоелементи. Зісканована інформація має растрову форму, програмне забезпечення сканера представляє неї в одному зі стандартних форматів, наприклад TIFF, GIF, PCX, JPEG, і для подальшої обробки може виконати векторизацію — переклад графічної інформації у векторну форму, наприклад, у формат DXF.

Для виведення інформації застосовують принтери і плотери. Перші з них орієнтовані на одержання документів малого формату (A3, A4), другі — для висновку графічної інформації на широкоформатні носії. У цих пристроях переважно використовується растровий (тобто порядковий) спосіб висновку зі струминною технологією печатки. Друкуюча система в струминних пристроях містить у собі картридж і голівку. Картридж — балон, заповнений чорнилом (у кольорових пристроях мається кілька картриджів, кожний з чорнилом свого кольору). Голівка — матриця із сопів, з яких дрібні чорнильні краплі надходять на носій. Фізичний принцип дії голівки термічний або п'єзоелектричний. При термодруку викидання крапель із сопла відбувається під дією його нагрівання, що викликає утворення пари і викидання крапельок під тиском. При п'єзоелектричному способі пропускання струму через п'єзоелемент приводить до зміни розміру сопла і викиданню краплі чорнила. Другий спосіб дорожче, але дозволяє одержати більш високоякісне зображення.

Дигитайзери, сканери, принтери, плотери можуть входити до складу АРМ або розділятися користувачами декількох робочих станцій у складі локальної обчислювальної мережі.

Особливості технічних засобів в АСУТП.

Специфічні вимоги пред'являють до обчислювальної апаратури, що працює в складі АСУТП у цехових умовах. Тут використовують як звичайні персональні комп'ютери, так і спеціалізовані програмувальні логічні контролери (ПЛК), названі *промисловими комп'ютерами*. Специфіка ПЛК — наявність декількох аналогових і цифрових портів, убудована інтерпретатор спеціалізованої мови, детерміновані затримки при обробці сигналів, що вимагають негайного реагування. Однак ПЛК на відміну від персональних комп'ютерів IBM PC розраховані на рішення обмеженого кола задач у силу спеціалізованості програмного забезпечення.

У цілому промислові комп'ютери мають наступні особливості: 1) робота в режимі реального часу (для промислових персональних комп'ютерів розроблені такі ОС реального часу, як OS-9, QNX, VRTX і ін.); 2) конструкція, пристосована для роботи ЕОМ у цехових умовах (підвищені вібрації, електромагнітні перешкоди, пилюка, перепади температур, іноді вибухонебезпечність); 3) можливість убудовування додаткових блоків керуючої, що реєструє, що сполучає апаратури, що крім спеціальних конструкторських рішень забезпечується використанням стандартних шин і збільшенням числа плат розширення; 4) автоматичний перезапуск комп'ютера у випадку "зависання" програми; 5) підвищені вимоги до надійності функціонування. Значною мірою спеціалізація промислових комп'ютерів визначається програмним забезпеченням. Конструктивно промисловий комп'ютер являє собою кошик (крейт) з декількома гніздами (слотами) для плат, що вбудовуються. Можливе використання мостів між крейтами. Як стандартні шини в даний час переважно використовуються шини VME-bus (Versabus Module Europe-bus) і PCI (Peripheral Component Interconnect).

VME-bus — системна шина для створення розподілених систем керування на основі устаткування, що вбудовуються, (процесори, нагромаджувачі, контролери введення-висновку). Являє собою розширення локальної шини комп'ютера на кілька гнізд об'єднаної плати (до 21 слота), можлива побудова багатомасерних систем, тобто систем, у яких ведучими можуть бути два або більш пристрої. Має 32-розрядні немультіплексорні шини даних і адрес, можливе використання мультіплексорні 64-розрядної шини. Пропускна здатність шини 320 Мбайт/с.

PCI — більш зручна шина для однопроцесорних архітектур, одержує усе більше поширення. Пропускна здатність до 264 Мбайт/з, розрядність шини 2x32 і (або) при мультіплексорні 64, архітектура з одним ведучим пристроєм. Мається ряд різновидів шини, наприклад шина CompactPCI, у якій уніфікований ряд геометричних і механічних параметрів.

Програмний зв'язок з апаратурою нижнього рівня (датчиками, виконавчими пристроями) відбувається через драйвери. Міжпрограмні зв'язки реалізуються через інтерфейси, подібні OLE. Для спрощення створення систем розроблений стандарт OPC (OLE for Process Control).

Рівні САПР

Технічне забезпечення сучасних САПР має ієрархічну структуру. Прийнято виділяти наступні рівні:

- - центральний обчислювальний комплекс (ЦВК),
- - автоматизовані робочі місця (АРМ),
- - комплекс периферійного програмно-керуючого устаткування.

Центральний обчислювальний комплекс призначений для рішення складних задач проектування. Являє собою ЕОМ середньої або високої продуктивності з типовим набором

систем

периферійних пристроїв. Можливе розширення цього набору деякими засобами обробки графічної інформації. Для підвищення продуктивності у ЦВК можуть використовуватися многопроцесорні або многомашинні комплекси.

Арми призначені для рішення порівняно нескладних задач і організації ефективного спілкування користувача САПР із комплексом технічних засобів. Включає у свій склад міні-ЕВМ і (або) мікро-ЕВМ, графічні і символні дисплеї, координатосъемщики, пристрою символного і графічного документування й інші з відповідним базовим і прикладним програмним забезпеченням. Для деяких АРМ характерний інтерактивний режим роботи з обробкою графічної інформації.

Комплекс периферійного програмно-керуючого устаткування призначений для одержання конструкторсько-технологічної документації і керуючих програм на машинних носіях для виконавчих технологічних автоматів. У його складі виконавче програмно-керуюче устаткування, засоби діалогової взаємодії. У складі ЕОМ з великим обсягом зовнішньої пам'яті. Подібні комплекси звичайно називають технологічними. На даному устаткуванні вирішуються задачі редагування, тиражування, архівного супроводу документації й ін.

Наявність зазначених рівнів приводить до відповідної структури програмного й інформаційного забезпечення САПР. У результаті рівні ЦВК, АРМ і ТК, спочатку виділювані як рівні технічного забезпечення, стають рівнями САПР.

Існуючі САПР поділяються на одно-, двох- і трьох-рівневі. В одноуровневих САПР, побудованих на основі ЦВК, виконуються процедури, що характеризуються високою трудомісткістю обчислень при порівняно малих обсягах вихідних даних. В одноуровневих САПР на основі АРМ виконуються процедури, у яких обсяги обчислень і документації, що випускається, порівняно невеликі. В одноуровневих технологічних комплексах зміст проектною документації визначається в результаті неавтоматизованого проектування, а виготовлення неї автоматизовано. При цьому обсяг документації, що випускається, може бути великим.

У дворівневих САПР можливі сполучення ЦВК-АРМ, ЦВК-ТК, АРМ-ТК.

Найбільшою мірою можливості автоматизованого проектування складних об'єктів реалізуються в трехуровневих САПР, що включають ЦВК, АРМ і ТК.

Територіальне рознесення окремих ЕОМ і комплексів САПР викликає необхідність включення до складу технічних засобів апаратури сполучення, передачі даних і телеобробки. При цьому технічні засоби великих САПР структурно поєднуються в обчислювальні мережі. Переваги організації обчислювальних мереж САПР полягають у наступному:

1. Користувачі, що працюють на апаратурі в конкретному підрозділі підприємства, одержують доступ до баз даних і програмних засобів, що мають у інших територіально рознесених вузлах обчислювальної мережі. Це розширює функціональні можливості САПР.

2. З'являється можливість оптимального розподілу навантаження між різними ЕОМ, а також можливість надання конкретному користувачеві в разі потреби значних обчислювальних ресурсів.

3. Підвищується надійність функціонування технічних засобів САПР.

Автоматизовані робочі місця проектувальників

Автоматизоване робоче місце (АРМ) проектувальника являє собою комплекс технічних засобів, що забезпечує оперативний і легкий доступ оператора до ЕОМ і допомагає реалізації ітераційних циклів проектування при діалоговому режимі роботи. АРМ дозволяє обмінюватися з ЕОМ інформацією в графічній формі. Функціонально АРМи можуть використовуватися як основу автономних САПР або підсистем функціонально-логічного, схемотехнічного, приборно-технологічного, конструкторського проектування різних САПР.

Комплекси АРМ можуть бути використані в якості:

- - одного з рівнів багаторівневих САПР,
- - робочих місць на рівні ЦОК,
- технологічних комплексів для адаптації конструкторського проекту до різного технологічного устаткування,
- - одного з технологічних маршрутів, включаючи спільну роботу з керуючими ЕОМ технологічного комплексу в режимі зворотного зв'язку,
- - інструментальних комплексів для розробки системного і прикладного програмного забезпечення для підсистем САПР.

Розглянемо два режими роботи АРМ: автономного і безпосереднього зв'язку з ЦОК.

1. В автономному режимі АРМ використовуються для рішення окремих проектних задач, що не вимагають високої продуктивності і великого обсягу оперативної пам'яті. Як правило, вони зв'язані з редагуванням графічної і текстової інформації і її документуванням.

Приклади проектних задач:

- проектування друкованих плат і механічних вузлів з випуском комплектів керуючих перфострічок і документації;
- проектування фотошаблонів мікросхем СВЧ вузлів і мікрополоскових трактів;
- підготовка керуючих перфострічок для верстатів з числовим програмним керуванням;
- проектування конструктивів.

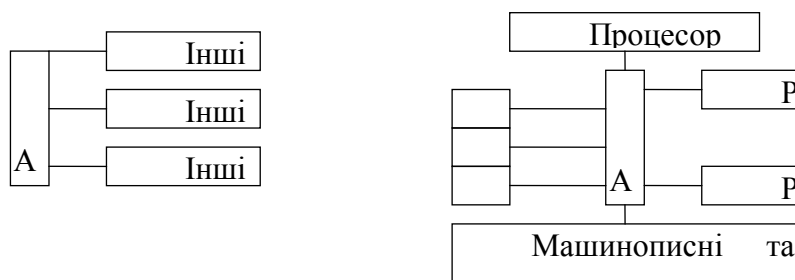
2. У режимі безпосереднього зв'язку з ЦОК технічні програмні засоби АРМ відіграють роль інтерактивного-графічного комплексу САПР і забезпечують виконання проектних операцій. Основне призначення АРМ у цьому випадку - забезпечення ефективного спілкування проектувальника з засобами автоматизації проектування.

Приклади проектних задач:

- - введення і редагування великих масивів вхідних даних і завдань;
- - керування режимами роботи САПР,
- - відображення і редагування результатів проектування;
- - випуск технічної документації;
- - моделювання й оптимізація елементів і схем БІС;
- - компонування і трасування плат друкованого монтажу і мікрозборок;
- - створення і поповнення банків даних.

Технічні засоби АРМ

В Армах технічні засоби групуються (див. рис.5) навколо високопродуктивної міні-ЕОМ. Вона зв'язана з периферійними пристроями, комплексами або іншими САПР каналами високої



А)



Рис.5. Структура технічних засобів АРМ (а) і склад апаратури робочих місць (б).

До каналів через інтерфейс типу "Загальна шина" підключаються технічні засоби робочих місць. Вони складаються з текстових і графічних засобів введення-висновку. Можливе використання ПК з оперативним і зовнішніми запам'ятовуючими пристроями. Склад технічних засобів АРМ для рішення задач проектування приведений у табл.

Таблица 4

Модель АРМ	Призначення	Склад технічних засобів
АРМ-Р-01	Мінімальний базовий комплект як основа інших варіантів	ЕОМ типу СМ, ОЗУ ємністю 8До слів, 16 розрядів, НМД 1370, дисплей VT-340
АРМ-Р-02	Розміщення, редагування графічної і текстової інформації, діалогу з ЦОК САПР на базі ЄС ЕОМ	АРМ-Р-01 із графічним дисплеєм ЭПГ-400
АРМ-Р-03	Інструментальний комплекс для розробки програмного забезпечення	АРМ-Р-01 із пристроєм мозаїчної печатки DZM-180

Комплексування АРМ

Комплектування технічних засобів САПР виробляється на основі наступних вимог:

- повноти,
- уніфікації,
- розширюваності,
- резервування,
- економічності розробки й експлуатації,
- експлуатаційної зручності і технологічності.

Повнота технічних засобів означає наявність у САПР набору технічних засобів усіх груп для виконання операцій по всьому циклі автоматизованого проектування.

Уніфікація технічних засобів означає використання однотипних одиниць устаткування для виконання тих самих функцій на різних рівнях САПР.

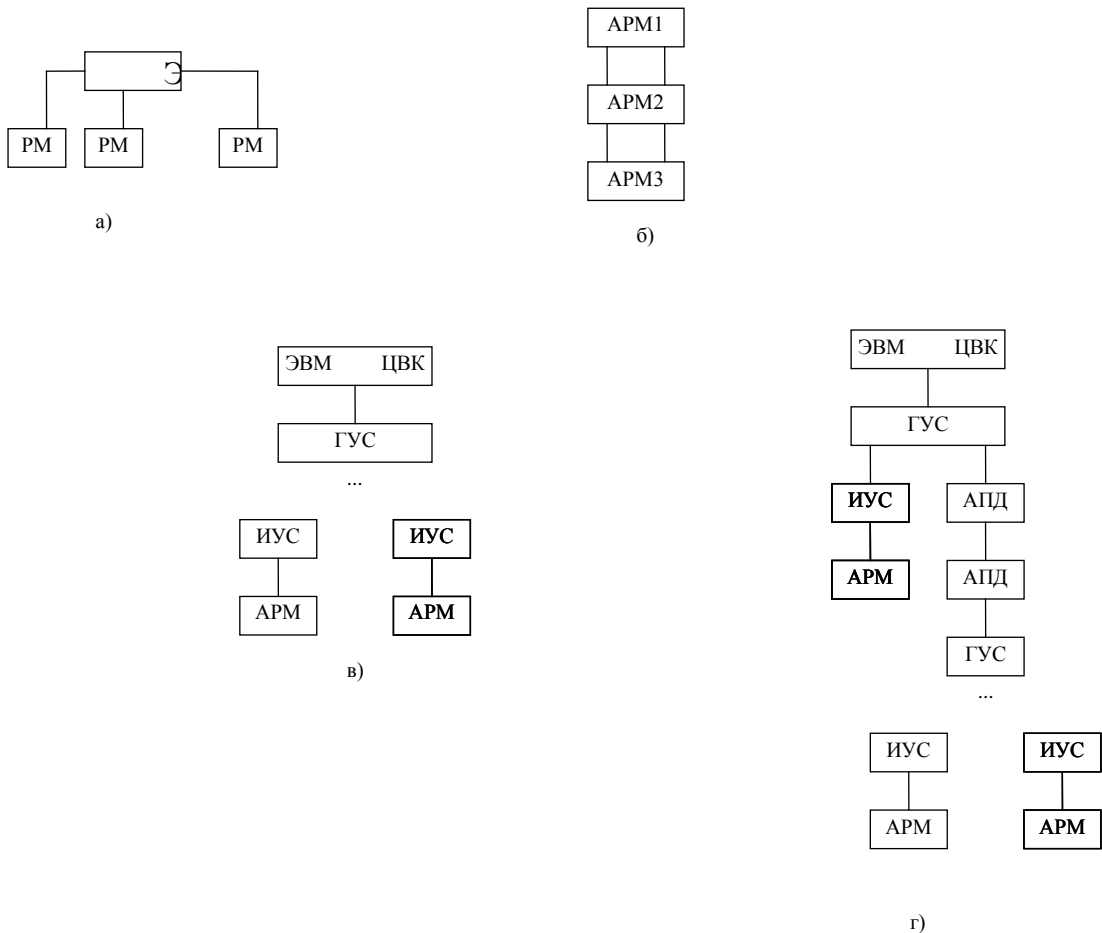


Рис.7. Варіанти використання АРМ у САПР:
а - групове АРМ; б - маршрутно-орієнтоване АРМ; в - ієрархічно зв'язані ЦВК-АРМ; г - вилучені АРМ в ієрархічних САПР.

Склад базового програмного забезпечення і технічних засобів АРМ дозволяє використовувати АРМ автономно або в декількох варіантах сполучення з іншою апаратурою.

Арми з одним або декількома робітниками місцями можуть поєднуватися в послідовний комплекс. Комплекс набудовується на рішення однієї або ряду послідовних на маршруті

проектування задач (рис.7.б).

Наприклад, введення завдання на розробку друкованих плат, синтаксичний контроль і розміщення елементів виконують на першому АРМ і по каналі зв'язку передають на другий АРМ. На ньому проводиться трасування, коректування і передача інформації для виготовлення комплексу конструкторської документації на третій АРМ.

Подібна маршрутно-спеціалізована конфігурація дозволяє усунути непродуктивні витрати часу на зміну магнітних носіїв, завантаження систем і обмежити кількість периферійних пристроїв.

Побудова інформаційно-графічних комплексів для САПР на базі АРМ (рис.7.а, б, у, г) розширює можливості і збільшує ефективність застосування САПР. НА ЦВК виконуються програмні модулі по директивах з Армов, а введення, контроль, висновок і редагування виробляється користувачем на засобах АРМ.

У цьому випадку потрібне вилучене розміщення технічних засобів АРМ і ЦВК. Для цього використовуються пристрої сполучення (групові пристрої сполучення - ГУС і індивідуальні ИУС), телефонні канали зв'язку, мультиплексори й апаратура передачі даних.

Розширюваність (відкритість) технічних засобів означає можливість кількісних і якісних змін у складі технічних засобів по зміні вимог до продуктивності і ступеня автоматизації проектування, а також появи нових більш зроблених типів устаткування.

Резервування технічних засобів реалізується дублюванням тих або інших засобів і дозволяє знизити вплив їхніх збоїв і відмовлень на функціонування САПР. Надлишок технічних засобів не тільки підвищує живучість САПР, але і є обов'язковою умовою успішної обробки потоку задач, інтенсивність якого змінюється в часі.

Економічність розробки технічних засобів дозволяє удешевити створення і впровадження САПР за рахунок послідовного многоетапного введення устаткування і нарощування потужності САПР із невеликим випередженням щодо росту поточних потреб.

Економічність експлуатації технічних засобів дозволяє знизити непродуктивні втрати за рахунок сполучення режимів реального часу з пакетною обробкою, колективного використання робочих місць.

Експлуатаційна зручність технічних засобів дозволяє збільшити продуктивність розроблювача і знизити рівень помилок при взаємодії оператора з ЕОМ за рахунок досконалості програмно-апаратного забезпечення.

Технологічність технічних засобів характеризується ступенем відповідності складу устаткування перелікові проектних операцій, властивих застосовуваній технології проектування заданого об'єкта.

Відповідно до цих вимог останнім часом визнані найбільш доцільними САПР з уніфікованих модулів. Вони мають досить розвиті технічні і базові програмні засоби. У складі периферійних модулів широке поширення одержали "інтелектуальні термінали" і "інженерні графічні станції", створювані на основі мікропроцесорних обчислювальних засобів.

Перспективи розвитку АРМ

Подальший розвиток АРМ зв'язаний з:

- використанням нових технічних засобів,
- розробкою нового базового і прикладного програмного забезпечення,
- розвитком технології автоматизованого проектування,
- об'єднанням АРМ в ієрархічні і мережні структури.

Широке поширення одержують професійні персональні ЕОМ з використанням процесорів 386 і 486 і спеціалізованих, а також відповідного програмного забезпечення.

Параметри нових АРМ:

- швидкодія - 5-10 млн.оп/сек,
- обсяг зовнішньої пам'яті - до 500 Мбайт,
- швидкість обміну інформацією з каналів зв'язку - понад 2 Мбайт/сек.

Робочі місця будуть обладнані кольоровими графічними дисплеями з регенерацією або растровими з розмірами екрана до 50 див по діагоналі і проєкційних із площею екранів у кілька квадратних метрів.

Базове програмне і лінгвістичне забезпечення повинні містити в собі:

- засобу створення багатомашинних мережних і ієрархічних структур;
- мониторну систему;
- операційну систему реального часу;

в'язаних один з одним АРМами по розробці логіки і схемотехники БІС, топології БІС і друковані плати, конструкцій вузлів і блоків виробів РЗА.

ЛІТЕРАТУРА

1. Гуцин, В. М. Управління розробками авіа та ракетно - космічних комплексів: Навчальний посібник .- М.: МАІ, 1999 .- 76с.
2. Івченко, Б. П., Мартищенко, Л. А. Монастирський, М. Л. Теоретичні основи інформаційно - статистичного аналізу складних систем .- М.: ИНФРА - М, 2002 .- 511с.
3. Норенков, І. П. Основи автоматизованого проектування / І. П. Норенков .- М.: Видавництво МГТУ ім. Баумана, 2000 .- 360С.
4. Половинкин, А. І. Основи інженерної творчості: Навчальний посібник / А. І. Половинкин - 3-е вид. Спб: Лань, 2007 .- 368с.
5. Схиртладзе, А. Г., Ярушин, С. Г. Проектування нестандартного обладнання: Підручник / А. Г. Схиртладзе, С. Г. Ярушин .- М.: Видавнича група «Нове знання», 2006 .- 424с.

Частина II.

Автоматизація пошукового конструювання в проектуванні СОС

10. Введення в пошукове проектування

Системи та їх проектування

Поняття системи, проекту і проектування

У математиці систему визначають як «сукупність елементів, пов'язаних між собою системо утворюючим відношенням R ». Іншими словами, система – це сукупність елементів, що перебувають між собою у певних взаємозв'язках і внаслідок цього набувають додаткові, системні властивості, які сам по собі не має жодний елемент системи. Оскільки техніка оперує штучно перетвореними явищами, необхідною і достатньою ознакою будь-якої технічної системи(ТС) буде наявність хоча б одного штучно створеного елемента. Одним з різновидів ТС, наприклад, є біотехнічна система, МЕМС і інші.

Кожен технічний об'єкт може бути за нашими бажаннями розглянути і як система і як не система. Відмінність полягає в тому, чи хочемо ми враховувати вплив компонентного складу системи і зв'язків між компонентами на властивості системи.

Слід зауважити, що якщо відносно поняття «система» існує деякий консенсус, то з поняттям «проекції» ситуація є неоднозначнішою.

У вітчизняній практиці під проектом найчастіше розуміють детально розроблений і документально оформлений план споруди чи конструкції; інколи проектом називають просто план вирішення проблеми.

Водночас тлумачний словник Вебтера визначає проект (project, від латинського projectus – «кинути уперед») як «будь-що, що задумують і планують, велике підприємство.» Словник з управління проектами «Dictionary of Project Managment» визначає проект як «діяльність, спрямовану на зміну будь-якої системи відповідно до поставлених цілей.» У «Кодексі знань про управління проектами», розробленому в 1987 році Інститутом управління проектами(Project Managment Institute, США), наводять таке визначення проекту: «Проект являє собою деяку задачу з певними вхідними даними і потрібними результатами(цілями), що зумовлюють їх розв'язання ». Коректніше визначення проекту наведено в роботі Моделювання евристичних задач проектування(Гліненко П.К. та ін.): «Проект включає в себе проблему, засоби її реалізації(розв'язання проблеми) і результати, що отримують в ході її реалізації.» Власне кажучи, проект являє собою логічно взаємопов'язану послідовність дій з досягнення попередньо поставленої мети, а простіше кажучи, проект – це те, що змінює наш світ. Зведення житлового будинку чи промислового об'єкта, програма науково-дослідних

Лекція 9. Технічне забезпечення процесу проектування робіт, реконструкція чи створення нового підприємства, розробка нової техніки, знімання фільму чи розвиток регіону – це все проекти.

Отже, у сучасній теорії і практиці проект у широкому розумінні процесу проектування тлумачать як цілеспрямовану зміну довільної системи через переведення її з одного стану в інший згідно з заданими обмеженнями і цілями; проект у вузькому розумінні – як план виконання робіт із здійснення цієї зміни. Відповідно під процесом проектування розуміють розроблення і реалізацію проекту.

Проектування довільної ТС складається з трьох основних етапів:

- Усвідомлення необхідності створення системи, стосовно визначення мети проекту;
- Абстрагування вхідних даних для проектування(технічних умов, технічного завдання), до складу яких входять параметри системи і обмеження до них;
- Власне проектування системи, яка б задовольняла обґрунтовані вхідні дані. Воно поділяється на етапи створення концепції проекту(фаза концептуального проектування), визначення і розв'язання основних задач проекту на рівні моделей, зокрема пошук адекватних технічних засобів цього розв'язання(фаза технічного проектування); перевірка отриманих результатів і коригування вибраних засобів(фаза тестування); планування і реалізація проекту(фаза планування і підготовки виробництва; виробництва та експлуатації; фаза утилізації).

В САПР та радіоелектроніці існує традиційне розуміння поняття проектування як створення такої сукупності описань нової чи вдосконаленої технічної системи(об'єкта(виробу) чи процесу), якої було б достатньо для виготовлення чи реалізації цієї системи в заданих умовах. Організаційний аспект проектування у такому разі відображається у блочно-ієрархічному підході до проектування, на підставі якого відбувається декомпозиції не стільки об'єкта проектування, але і самого проектного процесу, що дає змогу розподілити проектні роботи між окремими виконавцями. За такого розуміння проектування складається з комплексу дослідницьких, розрахункових і конструкторських робіт, виконання яких є необхідним і достатнім для отримання такої сукупності описань. Очевидно, що таке розуміння проектування є значно вужчим ніж це прийнято в межах, наприклад, концепцій Project Management, і в контексті останньої збігається з частинами етапів концептуального і технічного проектування. Проте саме на цих етапах якраз і проявляється специфіка проектування мікросистем. Саме на них зосереджена більшість інженерних задач і до 70% вибрані на проектування, тому більш детально розглянемо саме ці аспекти проектування.

Етапи концептуального та технічного проектування

Довільний проект починається з усвідомлення споживачем певної нереалізованої потреби, задоволення якої і становить мету проектування, та виникнення ідей її реалізації в умовах, що склалися. Цю початкову фазу життєвого циклу проекту і називають концептуальним проектуванням, зміст якого становлять такі етапи:

- Дослідження та описання ситуації, що обґрунтовує актуальність(необхідність чи доцільність) цього проекту, що містить детальне описання незадоволеної потреби та причини цього незадоволення; описання відомих способів її задоволення та причини їх незастосування;
- Розробка концепції проекту, тобто ідеї задоволення потреби у вигляді вербальної(словесний

систем

опис) моделі технічної системи, здатної задовільнити цю потребу. Створення такої системи – мета проекту, отже, на цьому етапі формують мету проекту;

- Формування технічних умов на ТС, визначення наявних ресурсів, обмежень, можливостей виготовлення ТС; наближена оцінка вартості, тривалості проекту кола фахівців тощо.

Після окреслення концепції проекту розпочинається фаза технічного проектування, де основні етапи:

- Формулювання проектних задач;
- Аналіз задач та визначення методів їх розв'язання;
- Розв'язання задач.

Проектні ситуації і задачі

Проект розпочинається з ідеї та формулювання мети і задач проекту, проходить фази підготовки, розробки і реалізації проекту аж до впровадження результатів у життя з отриманням суспільно корисного ефекту. Необхідність створення нової системи впливає переважно з деякої початкової потреби або бажання розширити можливості чи сферу використання наявної системи. (Деколи потребу можуть штучно створювати ті, хто буде її надалі задовільняти). Існує суперечність між можливостями наявної системи чи систем і тим, що ми хочемо отримати. Опис цієї суперечності, тобто реального стану справ, в якому фіксують незадоволеність деякій потребі, якщо неможливо задовільнити її наявними засобами, називають проектною ситуацією.

Отже, проектування починається з початкової проектною ситуації. Проектна ситуація тільки фіксує наявність певної потреби, яку неможливо задовільнити відомими засобами внаслідок їх принципової непридатності, або неможливості застосування в умовах, що склалися, і ставить вимогу задовільнити цю потребу як мету проектування. Проектна ситуація не передбачає ідеї реалізації цієї потреби і характеризується розпливчастим описанням умов.

Надалі на основі аналізу значущості різних елементів і функцій складають технічні умови на систему. Спочатку вони є неповними і у міру розробки проекту постійно уточнюються. На цьому етапі переходять від нечітко сформульованих описань до чітких задач і ідей. Саме на цьому етапі проектна ситуація розпадається на задачі, де під задачею розуміють постановку питання про отримання заданого результату(заданої функції із заданим рівнем якості її виконання) за заданих умов. Проектну ситуацію декомпонують на комплекс окремих задач, розв'язання кожної з яких, або їх комбінації забезпечує необхідне перетворення системи. Основу діяльності проектувальника на цьому етапі становить мистецтво правильного формулювання питань, логічний або інтуїтивний синтез(а не аналіз) цього комплексу проектних задач. Перехід від проектною ситуації до задачі – це творчий, евристичний крок. Його здійснення означає створення концепції проекту і перехід від концептуального до технічного проектування.

Результатом розв'язання проектною задачі є отримання технічних рішень, які забезпечують задоволення потреби(реалізацію потрібної функції з потрібними показниками якості).

Технічне рішення – це вказівка на засоби, які призначені для досягнення поставленої

Лекція 9. Технічне забезпечення процесу проектування

мети, де під засобами насамперед розуміють технічні засоби(пристрої, способи та речовини), що описують певними групами ознак. Отже, можна стверджувати, що технічним рішенням буде будь-яка технічна система, яка може реалізувати необхідну нам функцію. Тоді, використавши апарат факторного аналізу, проектну задачу можна декомпонувати на дві адитивні задачі: задачу знаходження множини систем з найкращим показником якості для певного критерію переваги. Задачі першого типу називають пошуковими, задачі другого типу – задачами вибору оптимальної структури чи оптимального синтезу, або просто задачами вибору. Розв'язуючи пошукову задачу, перебирають відомі фізичні і технічні об'єкти, щоб знайти такі їх поєднання, які б дали змогу реалізувати задану функцію в заданих умовах. Розв'язання задачі вибору передбачає знаходження оптимального варіанта реалізації функції за допомогою варіації елементів у межах множини однофункціональних систем.

Далі йде безпосереднє розв'язання задачі: пошук ідей розв'язку, розрахунок і оптимізація параметрів, коригування задачі і технічних умов до неї за результатами випробування, аналіз отриманих результатів, впровадження у виробництво, організація виробництва, розповсюдження і споживання одержаної продукції. Взагалі-то корегування будь-якого(і навіть всіх) попередніх кроків відбувається внаслідок аналізу результатів кожного етапу, тому під час проектування може бути багато ітераційних кроків внаслідок постійного порівняння отриманих результатів з метою.

Так виглядає загальний життєвий цикл будь-якого проекту, технічного, соціального чи економічного, а саме проектування являє собою процес покрокового прийняття рішень в умовах високого ступеня невизначеності. Цей процес декомпонують на етапи знаходження альтернатив, їх оцінки, оптимізації та вибору найкращої(за тим чи іншим критерієм) альтернативи. Кожний з етапів має власне методологічне забезпечення і виконується різноплановими фахівцями.

Інженерні задачі

Техніка пов'язана зі створенням систем, приладів чи процесів, які є чи здаються корисними для суспільства. Досягнення цих цілей називають інженерним проектуванням. Під час інженерного проектування переходять від початкової проектної ситуації, що фіксує факт непридатності старої технічної чи програмної системи для реалізації нових потреб, до формулювання комплексу інженерних задач, пошуку і знаходження методів їх розв'язання, впровадження результатів у життя. Інженерною задачею є задача з отриманням певного технічного результату, (певної нової або перетвореної технічної чи програмної системи), у заданих умовах. Розв'язання інженерних задач може вимагати як розрахунку чи зміни співвідношення параметрів наявної технічної чи програмної системи, так і зміни структури чи створення абсолютно нової(за структурою чи принципом дії) технічної чи програмної системи.

Отримання розв'язання інженерної задачі – це одержання нової(в якомусь плані) технічної чи програмної системи, і інженер – це передусім «проектувальник техніки(програм)». На жаль, цей першочерговий аспект інженерної діяльності загубився в різноманітті технічних дисциплін з велетенськими базами акумульованих спеціальних знань. Збільшення обсягу знань про природу і зростання складності задач, що розв'язувала техніка, призвели до виділення в техніці все вужчих спеціалізацій. У ході еволюції техніки роздрібнення технічних дисциплін є неминучим, і цей процес посилюється у міру накопичення

систем

знань у кожній галузі. Це змушує інженера-практика спеціалізуватися у все вузких областях(галузях). Ця спеціалізація, поряд з концентрацією розумових зусиль на глибинних механізмах явищ, має і певні негативні наслідки. По-перше вона до деякої міри відповідає за втрату сучасним інженером системного мислення. Спеціаліст, позбавлений системного сприйняття, нагадує сліпця з давньої індуської легенди: «Колись трьох сліпих підвели до слона і запропонували описати його. Один зі сліпців доторкнувся до ноги і сказав, що слон схожий на колону. Другий обмацав живіт і сказав, що слон – це велика стіна. Третій, який натрапив на хобот, порівняв слона з канатом.»

По-друге, ця «заспеціалізованість» призводить до того, що ми, наприклад, вважаємо інженером-електронником того, хто розбирається у роботі електронної схеми чи може намалювати структуру відомого радіопристрою, а не того, хто може спроектувати такі пристрої. Щоб бути інженером, недостатньо оволодіти сумою знань у певній галузі, треба на основі цих знань вміти отримувати нові знання, розробляти і реалізувати проекти, що змінюють світ. На жаль, досить часто система освіти сучасних спеціалістів спрямована, в основному, на засвоєння ними саме суми знань, як суми готових істин, а не на осягнення способів одержання цих істин. Водночас зі змісту інженерної діяльності очевидно, що людина, яка не володіє методами отримання нових знань, не може бути інженером. Опанування цих методів, які називають евристичними, утворює фундамент підготовки сучасного інженера.

Специфічна творча продуктивна діяльність зі створення нових технічних і програмних систем становить основний зміст роботи інженера. Оскільки, за факторологічним дослідженням, імовірність Q розробки проекту з фіксованим рівнем якості, прямо пропорційна до кількості проектувальників N , що працюють над одним проектом, середній продуктивності їх праці P , обсягу інвестицій у проектування(поетапно) I та потужності бази знань V і обернено пропорційна до середньої складності проектів C і часу на їх розробку T :

$$Q = \frac{NPIV}{CT}$$

Є очевидним, що підвищення продуктивності проектних робіт є найефективнішим напрямком розвитку перспективних галузей з високим ступенем інноваційності. Власне кажучи, інженерна діяльність – це і є інженерне проектування, і, як кожна інша діяльність, вона, на певному етапі свого розвитку, здійснюється за певними раніше розробленими правилами, за допомогою певних методів.

Проектні задачі: типи і методи розв'язання

На відміну від вченого, інженер не має свободи у виборі задач для розв'язання. Робота інженера – це розв'язання задач, які виникають під час пошуку та реалізації технічних рішень. Здебільшого отриманий розв'язок повинен задовільняти суперечливі критерії. Продуктивність коштує грошей і збільшення безпеки прозводить до зростання складності тощо. Якщо вимоги, що ставляться до задачі, не суперечить одна одній, то така задача є безконфліктною і її розв'язок отримують за допомогою звичайного інженерного розрахунку у межах певної галузі природничих чи технічних знань. Деколи такі задачі називаються задачами конструкторського чи технологічного розрахунку.

Лекція 9. Технічне забезпечення процесу проектування

В протилежному випадку коли задача полягає у знаходженні оптимального способу (методологічного, конструкторського, технологічного або організаційного) задовільняти певні суперечливі вимоги, її називають конфліктною. Типовий приклад конфліктної задачі – визначення опитальної форми корпусу корабля: корабель повинен бути якнайвужчим, щоб швидше рухатися (з погляду критерію очевидності), і якнайширшим, щоб не перевернутись (з погляду критеріїв стійкості та безпеки). Загалом можливі три варіанти розв'язання конфліктних задач:

1. компроміс між суперечливими вимогами (не дуже вузький і не дуже широкий корпус) з подальшою оптимізацією параметрів для отримання оптимального їх співвідношення;
2. нехтування однією з двох суперечливих вимог з одночасним максимальним задоволенням другої (дуже вузький і, відповідно, дуже швидкий і малостійкий корабель, як це, наприклад, віськові судна переслідування);
3. усунення суперечливих вимог за допомогою структурної перебудови системи (катамаран).

Розв'язок задачі, отриманий в результаті компромісу і оптимізації параметрів, називають оптимізаційним; розв'язок, що забезпечує усунення суперечливих вимог внаслідок структурної перебудови, з повною заміною типу структури або принципу дії системи включно, називають евристичним.

Конфліктні задачі допускають компромісне розв'язання в тих і тільки в тих випадках, коли існує принаймні одне значення параметра елемента, щодо якого ставлять конфліктні вимоги, яке б задовільняло обидві суперечливі вимоги. Якщо таких значень певна множина, то в межах цієї множини шукають оптимальне значення, тобто таке, яке б забезпечувало найкраще співвідношення конфліктних критеріїв. Це значення шукають методами оптимізації, тому як такі задачі, так і отримані розв'язки, часто називають оптимізаційними. Якщо ж не існує жодного значення параметра елемента, щодо якого ставлять конфліктні вимоги, яке б задовільняло обидві суперечливі вимоги, то таку конфліктну задачу називають евристичною. З погляду сучасної теорії множин довільна евристична задача є конфліктною задачею, для якої замкнені множини допустимих значень суперечливих критеріїв не мають жодної спільної точки. Елемент системи, стосовно параметрів якого постають суперечливі вимоги, запропоновано назвати проблемним елементом ξ , а самі задачі на усунення проблемного характеру усього елемента – проблемними. Розв'язок штатних задач може бути тільки евристичним і отриманий лише евристичними методами. Слід зауважити, що евристичний розв'язок допускають всі типи конфліктних задач, як ті, в яких можливий компроміс, так і ті, в яких це неможливо. Можна довести, стверджує в своїх роботах Г. С. Альтшуллер, що евристичний розв'язок буде найкращим навіть для компромісних задач.

Усунення «проблемності» проблемного елемента тотожне розв'язанню евристичної задачі і завжди вимагає хоча б мінімальної перебудови системи, тобто евристична задача завжди зводиться до задачі структурного синтезу. За ступенем дозволеності (чи реалізованості) перебудови вихідної системи під час розв'язання задачі евристичні задачі поділяють на три категорії:

- Задачі першого стану – задачі на усунення проблемного характеру елемента зміною чи заміною самого елемента з мінімальною зміною решти структури та системи та з повним збереженням її фізичного принципу дії;

систем

- Задачі другого стану – задачі на усунення проблемного характеру елемента за рахунок зміни структури всієї системи або підсистеми, до якої належить елемент, за повного збереження її фізичного принципу дії;
- Задачі третього типу – задачі на усунення проблемного характеру елемента зміною принципу дії системи. Ці задачі на синтез структури за заданою головною функцією F за відсутності жорстких обмежень на збереження всіх елементів колишньої структури, в яких системи-прототипу може і не існувати взагалі, тобто «проблемність» полягатиме саме у відсутності системи, здатної реалізувати необхідну функцію, а розв'язування задачі – у знаходженні принципу дії та дерева функцій елементів системи і побудові адекватної структури.

Відповідно до типів задач, розв'язання яких є метою проекту, в інженерному проектуванні розглядають декілька груп методів розв'язання задач:

- Методи розрахунку процесів та конструкцій, специфічні для кожної галузі техніки;
- Методи оптимізації процесів та конструкцій;
- Методи пошуку нових технічних розв'язань задач, які не вдається розв'язати спеціальними методами конструкторського розрахунку чи оптимізацією конфліктних параметрів. Це так звані евристичні, або пошукові методи; вони мають загальнотехнічний характер і ґрунтуються на різноманітних спробах інтенсифікувати і систематизувати генерацію нових ідей.

Методи перших двох груп детально вивчають студенти різних інженерних спеціальностей у відповідних спеціальних курсах. Евристичні методи, разом з методами аналізу проектної ситуації до задачі того чи іншого типу, становлять основу пошукового конструювання.

1.7. Підходи до розв'язання задач структурного синтезу

Предметний і функціональний підходи.

Ефективність проектування складних систем залежить не лише від наявних ресурсів і методів формулювання і розв'язання проектних задач, але і від способу вибору цих методів у кожному конкретному випадку, і який буде визначатися методологічним світоглядом проектувальника. Можливі дві вихідні позиції і два методологічні підходи до розв'язання проектних задач структурного синтезу. Перша позиція розглядає як мету проектування і результат розв'язання будь-якої проектної задачі створення чи вдосконалення певного об'єкту проектування, друга – реалізацію певної функції. Перший підхід називається предметним, другий – функціональним. Оскільки, як ми визначили раніше, причиною створення будь-якої системи є певна незадоволена потреба, нереалізована дія, то не викликає сумніву, що саме функціональний підхід здатний забезпечити максимальну ефективність проектування.

Раціоналізаторський та проектний підходи.

У межах як предметного, так і функціонального підходів теоретично можливі два методологічні підходи до розв'язання проектних задач. Перший забезпечує перехід від

Лекція 9. Технічне забезпечення процесу проектування наявного стану системи до якогось кращого стану (раціоналізаторський підхід), другий – перехід від ідеального стану до реалізованого (проектний підхід). Сучасна концепція проектування схиляється до другого з цих підходів. Ця концепція є орієнтованою на користувача, вона вважає вимоги майбутнього користувача безумовними. Згідно з моделлю Норіка (K.W. Norris; The morphological approach to engineering design II Conference for Design Methods, Oxford Pergamon Press, 1963), формулювання задачі, тобто інтерпретація потреби мовою проектувальника поширюється на усе проектування. Це значно розширює межі концептуальної частини під час проектування за рахунок звуження рутинної частини. Об'єднання цих двох методологічних підходів утворює метод формування розвитку системи. (В.Гашарский).

Морфологічний та трансформаційний підходи.

Методи традиційного розв'язання евристичних задач перебором варіантів на основі досвіду та інтуїції проектувальника виявилися неефективними у разі складних систем і поступилися методам генерації розв'язків на основі фрагментів альтернатив (морфологічні методи (Одрин В. М., Картавов С. С., Морфологический синтез систем. – К.: Наукова думка, 1977. – 148 с.)) або на основі трансформації прототипу (трансформаційні методи) (Гліненко А. К. та ін. Моделювання евристичних задач проектування – Львів: Телемаркет, 1986).

Слід зауважити, що з погляду теорії прийняття рішень довільне рішення E_i являє собою вибір одного з деякої множини варіантів розв'язань $E_i \leq E$. Якщо множина $E = \{E_1, E_2, \dots, E_i, \dots, E_n\}$ скінченна і зліченна, тобто існує скінченна кількість варіантів розв'язання, то задача прийняття рішення є комбінаторною задачею кінцевого вибору. Тобто вибору на скінченній множині альтернатив. Такий вибір найкращого розв'язку ведеться за приписуваням йому результатом e_i за допомогою кількісного критерію $E_0 = \{E_i | E_i \leq E \wedge e_i = \max e\}$. Методи розв'язання задач кінцевого вибору до яких належать задачі оцінки і вибору оптимальної альтернативи, достатньо формалізовані та автоматизовані.

Якщо ж множина E можливих розв'язків нескінченна, тобто кількість варіантів розв'язків до кінця невідома, то ми маємо справу із задачею нескінченного вибору (вибору на нескінченній множині рішень). Саме до таких задач належать евристичні пошукові задачі генерації альтернатив. Оцінюють розв'язки таких задач переважно за номінальною шкалою; критеріям належності рішення E_i до множини можливих (допустимих) рішень E є принципова здатність цього рішення взагалі розв'язати задачу, без кількісної оцінки результату. Пошук таких рішень, що полягає у знаходженні нових альтернатив, являє собою вищого ступеня складності, розв'язання яко до кінця не піддається формалізації.

Згідно із сучасним уявленням загальний спосіб розв'язання задач з вищим ступенем складності і новітності полягає у заміні їх на задачі меншої складності і новітності. Для задач прийняття рішень цей спосіб реалізується у знаходженні розв'язань задач нескінченного вибору через звернення їх до задач кінцевого вибору, що вимагає створення скінченної зліченної множини допустимих рішень у вигляді бази даних стандартизованих альтернатив, структурованої за певними ключовими ознаками пошукових задач з подальшим створенням відповідного трансформаційного або перебірного алгоритму вибору одної з цих альтернатив. За морфологічного підходу зліченна множина допустимих рішень складається з множини варіантів готових рішень чи їх фрагментів, за трансформаційного – з множини готових способів перетворень (евристичних прийомів) чи способів реалізації дії (ефектів).

Морфологічний підхід ґрунтується на комбінаторному принципі пошуку рішення на морфологічній множині варіантів готових рішень, або її фрагментів, з яких рішення синтезують тим чи іншим способом. Наприклад, в галузі РЕА метод успішно застосований для синтезу швидкісних детекторів сигналів і для створення електронно-оптичних здавачів лінійних переміщень. Основою трансформаційного підходу є послідовна трансформація (перетворення) ТС – прототипу за допомогою евристичних прийомів, під якими розуміють уніфіковані способи перетворення або техн. систем ао уявлень про них у свідомості людини, які б забезпечили покращення показників якості системи.

Пошукові методи: класифікація та способи підвищення ефективності

Евристичні (пошукові) методи можна класифікувати за низкою ознак, зокрема:

- - за сферою придатності (типом розв'язуваних пошукових задач) та етапом проектування;
- - за механізмом підвищення ефективності та інструментами їх реалізації;
- - за наявністю критерію переваги;
- - за спрямованістю;
- - за ступенем алгоритмізації, базових знань, тощо.

Сфера придатності.

За типом розв'язуваних задач пошукові методи поділяються на методи подолання недоліків відомих технічних об'єктів, які не вдалося подолати алгоритмічними методами (задачі першого та другого типів), і методи пошуку нового за принципом дії об'єкту (задачі третього типу). За місцем у проектуванні виділяють методи аналізу та декомпозиції проектної ситуації; методи її перетворення; методи генерації ідей рішень та комплексні методи, які складаються у комбінації методів, що були застосовані на різних етапах.

Механізми підвищення ефективності.

Евристичні методи, які використовують сьогодні в інженерній практиці, ґрунтуються на різних варіантах підвищення ефективності природнього методу перебору варіантів. Аналіз пошуку нових рішень за цим методом дає змогу стверджувати, що інтенсивність пошуку рішень можна збільшити за допомогою:

- -впливу на інтенсивність генерації ідей за рахунок збільшення мотивації отримання нової ідеї певними фізіологічними чи організаційно-економічними методами;
- -розширення поля пошуку за рахунок збільшення хаотичності пошуку, це так звані асоціативні методи, найвідомішими з яких є методи колективного пошуку на базі мозкового штурму та використання механізму аналогій;
- -систематизації перебору варіантів за допомогою різноманітних варіантів морфологічних таблиць;
- -логічної алгоритмізації пошуку, тобто пошуку у формі відповідей та поставлені у певний спосіб запитання. Характер цих запитань і їх послідовність не є випадковими, а ґрунтуються на закономірностях розвитку техніки. Унаслідок цього саме ця група методів

Лекція 9. Технічне забезпечення процесу проектування забезпечує найбільшу ефективність пошуку нових ідей.

Підтримання механізмів збільшення ефективності пошукових методів можливе за рахунок відповідної організації роботи розробників, впливу на психологічний чи фізіологічний стан людини; систематизацією, уніфікацією, стандартизацією множини альтернатив чи способів їх знаходження та алгоритмізацією процедури пошуку. Відповідно до основного задіяного засобу збільшення ефективності методи ефективізації пошуку поділяють на організаційні, фізіологічні, психологічні, систематичні (морфологічні) та алгоритмічні.

Критерії переваги.

Під критерієм переваги розуміють правило, на основі якого одну з однофункціональних систем вважають кращою за іншу [Гушкин А. С. Оптимізація радиоелектронних у-в], тобто умови, за яких $K_1 > K_2$. Наявність критерію переваги дає змогу здійснити об'єктивний вибір "найкращого" рішення.

Спрямованість.

Спрямованість θ методу пошуку характеризує трудомісткість пошуку і визначається співвідношенням кількості одержаних однофункціональних систем з кращим ніж у вихідної (початкової), показником якості N_i до загальної кількості отриманих при пошуковій однофункціональних систем N [Глазунов В. Н. Параметрический метод разрешения противоречий в технике].

$$\theta = N_i/N$$

Розвинутість методів.

Розвинутість пошукового методу характеризується його придатністю до автоматизації, який визначають ступенем алгоритмізації методу, можливістю адаптації алгоритму до розв'язання задач різного типу, розвинутістю і зручністю для користувача бази знань методу. Ефективність методу пошуку концепції розв'язання оцінюють кількістю спроб, які необхідні для отримання найкращого результату, що прямо залежить від наявності критерію переваги і спрямованості методу. Загальновизнаним нині є те, що за всіма цими ознаками найефективнішими є трансформаційні методи логічної алгоритмізації пошуку.

Найпотужніші бази евристик і найрозвинутіший алгоритм їх застосування мають трансформаційні методи на основі теорії розв'язання евристичних задач [Альгшуллер Г.С. Творчество как точная наука] та її модифікацій, що зумовлює ефективність їх використання на різних етапах структурного синтезу ТС різних класів.

11. Комп'ютерні методи пошукового конструювання

Аналіз функцій технічних систем

Опис функцій технічних систем і їх елементів. Визначення основних понять

Методи автоматизованого конструювання фізичних принципів дії і технічних рішень, що будуть розглядатися, в значній мірі побудовані на поділі технічних систем (ТС) на елементи, які мають певні функції, і пошуку засобів реалізації цих функцій. У зв'язку з великою роллю функціональної основи в пошуковому конструюванні розроблена методика визначення функцій ТС та їх елементів з встановленням між ними функціональних взаємозв'язків. До технічних систем будемо відносити різні складні об'єкти проектування, включно з мікросистемами, засоби праці, обробляючі транспортні та енергетичні машини, прилади, автомати, вироби військової техніки, а також елементи цих об'єктів.

Не дивлячись на широке використання понять функцій ТС або їх елементів в методиках пошукового конструювання, функціонально-вартісного аналізу, в патентознавстві та інших сферах інженерної діяльності, самі ці поняття чітко не визначені. Тому більшість розробників формулюють і описують функцію ТС і її елементів інтуїтивно. Це значить, що опис функції одного і того ж класу ТС і для одної і тої ж мети у різних розробників може суттєво відрізнятись і мати різну повноту і точність.

В літературі і на практиці звичайно використовують дві форми опису функцій. Одна побудована на моделі «чорного ящика», коли під функцією ТС розуміють дію для перетворення розглянутою ТС деякого вхідного або початкового стану (ситуації) N в бажаний кінцевий результат R. Структурна формула такого представлення функції має вигляд

$$F = (N \rightarrow R), \quad (1)$$

де F – функція розглянутої ТС; знак « \rightarrow » вказує на перетворення початкового стану N в кінцевий результат R. Опис функцій по цій формі, наприклад, для електрокаміна має вигляд

$$F_e = (\text{«холодна кімната»} \rightarrow \text{«тепла кімната»}).$$

Іноді крім станів N і R необхідно ще описати особливі умови і обмеження H, що накладаються на реалізацію функції. При цьому формула (1) буде мати вигляд

$$F = ((N \rightarrow R), H) \quad (2)$$

Для розглянутого прикладу H = (при використанні електричного струму).

Другу форму можна назвати природною формою опису функції на мові спеціалістів. Наведений приклад опису функції по формулі (1) в другій формі буде мати наступний вигляд:

$$F_e' = (\text{«обігрів кімнати електричним струмом»}).$$

Візьмем за основу другу форму опису функції, яка більш зручна і конкретна для людини і машинної обробки інформації по функціях ТС. Крім того, друга форма більш універсальна. Наприклад, таку функцію як «зберігання рідини в резервуарі», важкувато описати через першу форму.

Серед різних авторів, що давали рекомендації по природній формі опису функції ТС, заслуговує на увагу представлення, викладене в роботах Ф.Ханзена: «Всякому технічному

Лекція 9. Технічне забезпечення процесу проектування

пристрою, що складається з деякої кількості елементів, присущий певний зв'язок між ними, який визначається функцією пристрою. Ця функція, з яких би часткових функцій вона не складалась, характеризується однозначною метою. Остання, як правило, може бути описана невеликою кількістю слів і має назву ядра або цілі функції. Для того ж, щоб охопити функцію в цілому, необхідно поряд з ціллю функції знайти ще й обмежуючі умови.» Таким чином Ф.Ханзен виділяє дві компоненти в описі функції, однак чітких рекомендацій по їх опису він не дає. В зв'язку з цим, професором Половінкіним О.І. зроблена спроба сформулювати рекомендації по найбільш повному, точному і однозначному опису функцій ТС і її елементів.

Природню форму опису функції ТС формалізовано можна представити як трійку компонент:

$$\mathbf{F} = (\mathbf{D}, \mathbf{G}, \mathbf{H}), \quad (3)$$

де \mathbf{D} – вказівник дії розглянутої ТС, що приводить до бажаного результату; \mathbf{G} – вказівник об'єкта (об'єктів), на який направлено цю дію; \mathbf{H} – вказівник особливих умов і обмежень, при яких виконується дія.

Наприклад:

ТС: електрокамін

\mathbf{D} : отримання теплоти

\mathbf{G} : з електричного струму

\mathbf{H} : відсутні (–)

Повне речення опису функції ТС може мати наступну граматичну форму:

«Призначення (Т) – (F)» або «Призначення (Т) – (D,G,H)», де (Т) – найменування ТС; (F) – опис функції по формулі (3), наприклад, «призначення мембрани – передача коливань».

Опис функції по формулі (3) можна перетворити в опис по формулі (1) або (2). При цьому може бути використаний один з двох способів перетворення. В першому способі ситуацію \mathbf{N} описують, вказавши дію \mathbf{D} . В другому способі ситуацію \mathbf{N} описують, вказавши об'єкт \mathbf{G} і початковий стан дії \mathbf{D} , а кінцевий результат \mathbf{R} , також вказавши об'єкт \mathbf{G} і кінцевий стан дії \mathbf{D} . Компоненту \mathbf{H} описують однаково.

Будь-яка сформульована функція відповідає деякому класу ТС, що реалізують цю функцію або (і) можуть її реалізувати. Причому опис функції може відповідати більш або менш широкому класу ТС. Найбільш широкому класу ТС відповідає такий опис функції, в якому компоненти \mathbf{D} і \mathbf{G} мають найбільш узагальнений вираз, а компонента \mathbf{H} відсутня. Прикладом такої функції може служити «розділення твердих тіл». При конкретизації описів компонент \mathbf{D} і \mathbf{G} , при збільшенні кількості умов і обмежень \mathbf{H} і при більшій їх конкретизації опис функції буде відповідати різним більш вузьким класам ТС. Це легко побачити при заміні в приведеному прикладі дії «розділення» на більш конкретні (різання, розрубання, пиляння, розділення лазерним променем тощо) і при заміні об'єкта «тверді тіла» більш конкретними (тверді тіла з деревини, скла, сталі або ще більш конкретно: зі сталі – 3 або інструментальної сталі) і при введенні таких умов і обмежень: при температурі +300°C або -50°C, товщиною 1 мм або 1000 мм, під водою на глибині 5 м або 5000 м.

Тому при описі функції завжди мають на увазі клас ТС, якому повинен відповідати цей опис. В свою чергу широта класу ТС залежить від конкретних задач, в яких використовується опис функції. Наприклад, при інженерному прогнозуванні звичайно мають на увазі досить широкий клас, при проектуванні – значно більш вузький, а при пошуку несправностей – клас

конкретних однакових ТС.

В цілому опис функції необхідно формулювати конкретніше, коротше і простіше і не пробувати в опис функції включати ТЗ на проектування, яке окрім функції ТС включає ще список спеціальних вимог.

В основу аналізу функцій ТС покладений принцип виділення і розгляду структур з дворівневою ієрархією: будь-яку ТС можна розділити на декілька конструктивних елементів, що мають цілком визначену функцію по забезпеченню роботи ТС або її елементів. При цьому розглянута ТС представляє собою верхній рівень, а виділені елементи – нижній. Якщо необхідно продовжити (поглибити) аналіз, то кожний з виділених елементів нижнього рівня розглядається як самостійна ТС, яку також можна розділити на декілька конструктивних елементів і т.д. Об'єднання таких структур з дворівневою ієрархією дозволяє отримати багаторівневу ієрархічну структуру.

Іноді замість поняття «конструктивний елемент» використовується поняття «функціональний елемент», оскільки можливим є такий поділ ТС на конструктивні елементи, при якому так «розрізаються» функціональні елементи, що вони не можуть виконувати свою функцію.

Під **функціональними елементами ТС** будемо розуміти вузли, деталі і частини деталей, включно з неподільними елементами.

Вузлом ТС будемо називати множину конструктивно зв'язаних деталей, виконуючих у сукупності хоча б одну функцію по забезпеченню роботи інших функціональних елементів розглянутої ТС або самої ТС.

Зауважимо, що крупні складні вузли часто можна розділити на більш прості функціональні підвузли, які в свою чергу можна розділити ще на більш прості підвузли, що складаються з декількох деталей і т.д.

Деталлю ТС будемо називати окреме тіло з однорідного матеріалу, яке виконує хоча б одну функцію по забезпеченню роботи інших функціональних елементів ТС або самої ТС.

Деталі можуть бути виконані з твердих, пружних, пластичних, рідких матеріалів або газів. До деталей відносяться, наприклад, гумова камера, повітря і деталі нипельного вузла деяких коліс автомобіля (в безкамерних замість камери - покривка), струя води в гідромоніторі, полум'я газової горілки і т.і. До деталей не відносять матеріали при обробці, якщо вони одночасно не виконують додаткових функцій по забезпеченню роботи ТС. Деталі при послідовному функціональному поділі на частини можна в результаті «розрізати» на неподільні елементи, що не допускають подальшого функціонального поділу.

Неподільним елементом будемо називати частину деталі (або самі деталі), яка має мінімальну кількість (але не менше одної) функцій по забезпеченню роботи інших елементів, при будь-якому діленні (поділі), якої з'являються елементи, що не мають самостійної функції або з однаковими функціями.

До неподільних елементів шарікопідшипника, наприклад, можна віднести такі конструктивні елементи як шаріки, робочу поверхню зовнішнього кільця і т.і. Поряд з тим рідина в рідинних мікросистемах є одночасно деталлю і неподільним елементом.

Синтез фізичних принципів дії технічних систем

Кількісний синтез фізичних принципів дії (ФПД)

Якісний синтез ФПД представляє собою доведення тільки необхідних умов реалізації заданої функції ТС або її елемента. З іншої сторони, вибір або синтез ФПД – це одна з підзадач в розробці нових ТС, коли за вибором ФПД необхідно розв'язувати задачу вибору або синтезу технічного рішення (ТР). При пошуку ТР прагнуть задовільнити деякий список вимог – (ТЗ) технічне завдання. Кожна вимога в цьому списку представляє собою найменування якогось обмеження, що накладається на ТС або її елемент зі вказанням необхідних значень. При цьому деякі вимоги мають критеріальний характер, тобто по них бажано мати найбільше або найменше значення.

Тому мета кількісного синтезу ФПД – отримання таких ФПД, які б в найбільшій мірі задовільняли вимогам ТЗ. Інакше кажучи, чим краще буде пророблено необхідне рішення на стадії кількісного синтезу ФПД, тим менше необхідно буде зусиль при отриманні допустимих і найкращих ТР. При якісному синтезі ФПД іноді отримують дуже велику кількість рекомендованих ФПД, які конструктор повинен аналізувати і доробляти для вибору найбільш прийняттого рішення. При цьому велика кількість (більшість) ФПД, знайдених шляхом якісного синтезу, по ряду обмежень є неприпустимими.

Для кількісного синтезу ФПД в банку даних повинні знаходитись більш детальні описи фізичних ефектів (ФЕ). Вибір складу інформації для більш детального або кількісного опису ФЕ, яка б в найбільшій мірі задовільняла вказаній меті кількісного синтезу ФПД, представляє собою досить нелегку задачу. Справа в тому, що обсяг цієї інформації для опису окремого ФЕ дуже швидко зростає і в зв'язку з цим стає проблематичним створення достатньо компактного інформаційного масиву ФЕ. Коротше кажучи, розробка стандартної форми кількісного опису ФЕ для створення відповідного найбільш повного інформаційного масиву колективного користування є досить відповідальною задачею, яку необхідно розв'язувати. Наведемо деякі рекомендації по кількісному опису ФЕ.

Аналіз ТР різних ТС і ТЗ вказує на доцільність використання при їх розробці наступних характеристик, що доповнюють якісний опис ФЕ.

Математична модель ФЕ може бути представлена у вигляді функції

$$C = f(A, B, P_b),$$

де A – вхідні впливи, B – фізичні об'єкти, P_b – зовнішні впливи, які змінюють значення вихідного ефекту C .

Крім цього математична модель повинна вказувати критичні зовнішні впливи P_k , при яких розглянутий ФЕ припиняє свою дію. В основі математичної моделі лежать аналітичні залежності, апроксимуючі формули, таблиці тощо. В остаточному вигляді в банку даних по ФЕ математична модель повинна представляти собою програмний модуль.

Якщо ФЕ є виділеним шляхом розщеплення зістикovanого ФЕ, то його математична модель містить перелік інших результатів впливу C_2, \dots, C_k і відповідні їм ФЕ. Можуть бути також присутні спеціальні вказівки по моделюванню в цілому здійснюваного ФЕ.

Характеристика фізичних об'єктів B . Кожний об'єкт у відповідному ФЕ містить основні фізичні характеристики, необхідні для використання математичної моделі і розробки ФПД і ТР.

Графічні характеристики фізичного ефекту включають, як правило, два типи зображень: схематичне представлення ФЕ, яке суттєво доповнює і пояснює опис сутності ФЕ або його матем. модель, і один або кілька прикладів конструктивних схем (спрощених ТР) практичного використання ФЕ.

Енергетична характеристика ФЕ, - описують відношенням, аналогічним ККД, $\theta = E_C/E_A$, де E_A , E_C – опис А і С на енергетичному рівні, тобто це кількість енергії на вході і виході. Вочевидь, що $0 < \theta < 1$ і θ залежить від В і Р_б.

Часова характеристика ФЕ t(A → C) дорівнює часу перетворення А в С і також залежить від В і Р_б.

Слід зауважити, що графічна, енергетична і часова характеристики мають сенс не для всіх ФЕ і відповідно для деяких ФЕ будуть відсутні. Вкл. цих характеристик може бути викликано тим, що вони можуть представляти певний інтерес прОи розробці ПФД, а потім і ТР. Енергетична і часова характеристики також представлені відповідними програмними модулями.

Можна також зауважити, що інформація по кількісному опису для різних ФЕ буде нерівноцінною. Тобто в описі деяких ФЕ будуть міститись, наприклад, досить грубі або малодостовірні математичні моделі, або їх взагалі може не бути, оскільки банк даних відбиває сучасний стан в розвитку окремих розділів фізики і хімії, а також ступінь вивченості окремих ФЕ. Стосовно різних кількісних і графічних характеристик ФЕ, його об'єктів В, то вони повинні бути такими, щоб банк даних можна було використати при пошуковому конструюванні більшості ТС, тобто кількісний опис ФЕ повинен бути в найбільшій мірі інваріантним до класу ТС.

Основна суть кількісного синтезу ФПД полягає у виборі найбільш ефективних допустимих рішень з множини ФПД, отриманих на стадії якісного синтезу і деякої їх наступної доробки. В основу алгоритму кількісного синтезу ФПД покладені наступні процедури.

1.Перевірка відповідності «входів» і «виходів»:

$$[A_1] \cap F_{вх} \neq \emptyset; [C_n] \cap F_{вих} \neq \emptyset. \quad (4)$$

де $[A_1]$, $[C_n]$ – множини значень відповідно початкового вхідного впливу A_1 і кінцевого результату C_n ; $F_{вх}$, $F_{вих}$ – множини значень, що описують функцію ТС, яка відноситься до одної з головних вимог ТЗ. Множини значень можуть бути і точковими (тобто з одного елемента). Після перевірки відповідності входів і виходів залишається множина ФПД, які задовільняють обмеженню (4), і в кожному такому ФПД залишаються тільки допустимі набори фізичних об'єктів відповідно \bar{B}_1 і \bar{B}_n .

2.Перевірка кількісної сумісності ФЕ:

$$[C_i] \cap [A_{i+1}] = \emptyset \quad (i = 1, \dots, n-1),$$

де $[C_i]$, $[A_{i+1}]$ – множини значень фізичних величин для обох наборів фізичних об'єктів \bar{B}_i , \bar{B}_{i+1} . В результаті цієї перевірки залишаються тільки ФПД з кількісно сумісними ФЕ, і при цьому для кожного ФЕ вибирають лише фізичні об'єкти, що забезпечують таку сумісність.

3.Перевірка обмежень на речовини і матеріали:

$$B \cap B_s = \emptyset \text{ або } B = B_r, \quad (5)$$

де B – множина всіх фізичних об'єктів у вигляді речовин і матеріалів, які використовуються у всіх ФПД, що розглядаються; B_s – множина речовин і матеріалів. По умові (5) відсікаються ФПД, що залишилися після процедури 2.

4.Критеріальний вибір ФПД.

З ТЗ вибирають такі вимоги T_j , значення яких можна визначити (наприклад, розрахувати) для кожного (або більшості) ФЕ в структурі ФПД.

Для цих вимог можна побудувати критерій у вигляді суми (по всіх ФЕ), добутку або іншої функції величин, які цікавлять. До вимог T_j можуть відноситись, наприклад, ККД, час спрацьовування, тепловиділення, енерговитрати, маса і т.і.

5.Перевірка впливу зовнішнього середовища.

З описів математичних моделей ФЕ для кожного ФПД складають список можливих зовнішніх впливів $U, P_b, P_{кр}$, що впливають на роботу ФПД. Далі на основі аналізу ТЗ із списку можливих зовнішніх впливів виділяють і розглядають впливи, які мають місце, тобто розв'язують одну з задач фізичного аналізу, в результаті чого вибирають найкращі працездатні варіанти ФПД.

6.Перевірка внутрішніх фізичних впливів у структурі ФПД.

Для кожного вибраного ФПД розв'язують задачу фізичного аналізу з використанням описів математичних моделей ФЕ, що дає додаткову інформацію для вибору найкращого (найкращих) ФПД.

Інформаційне забезпечення синтезу технічних рішень

Машинний синтез технічних рішень(ТР) здійснюється, в основному, за допомогою трьох інформаційних масивів: масиву відомих ТР; масиву вимог до відомих і нових ТР.

Представлення окремого технічного рішення у вигляді ієрархічного дерева

Люба технічна система(ТС) складається з ряду взаємопов'язаних функціональних елементів, що утворюють деяку ієрархічну структуру. Більш детальний інженерно-технічний опис цієї структури в патентознавстві та інших сферах інженерної практики називають ТР. Розглянемо зміст цього поняття для наступного використання.

Технічним рішенням будемо називати конструктивний опис функціональної структури ТС, включно з інформацією про функціональні елементи (блоки, вузли, деталі), їх взаємне розташування і взаємозв'язки, особливості конструктивного виконання, геометричну форму, конструкційні матеріали та інші суттєві ознаки, включно з принципово важливими співвідношеннями значень параметрів.

У відповідності з цим визначенням для опис самих різних ТР використовують наступні групи конструктивних ознак ТР, зміна яких, як правило, забезпечує дискретний перехід від одного рішення до іншого.

1. Функціональні елементи звичайно утворюють головні конструктивні ознаки. Кожний елемент виконує одну або кілька функцій по забезпеченню роботи взаємопов'язаних з них елементів або ТС в цілому. Для опису функціональних елементів використовують загальноживані назви деталей, вузлів, механізмів і т. д. Часто найменування функціонального елемента співпадає з функцією, яку він виконує (наприклад, двигун, нагрівач, фільтр, сепаратор).

2. Взаємне розташування елементів характеризує розташування функціональних елементів відносно одне одного в просторі, визначає конструкцію (конструктивну схему) ТС в цілому. Для опису ознак взаємного розташування найчастіше використовуються терміни «розміщений», «виявлений», «розташований» і т. д. разом з назвами функціональних елементів, по відношенню до яких визначається положення даного елемента.

3. Взаємозв'язок елементів. Ця група ознак характеризує такий взаємний зв'язок елементів, який забезпечує їх функціональну єдність в роботі ТС. Зв'язки можуть бути самі різні (наприклад, шарнірне з'єднання, електропривід, лазерний промінь і т.д.). В описі ознак зв'язку вказують елементи, між якими є зв'язок, і засоби за допомогою яких цей зв'язок здійснюється.

4. Особливості конструктивного виконання елементів використовуються для їх укрупненої характеристики при описі ТС. Ці ознаки дозволяють згрупувати ТС, наприклад, по фізичному принципу дії (п'єзоелектричний динамометр), або по виду енергії, яка використовується для функціонування ТС (електрочайник), або по технологічному способу виготовлення (литий корпус) і т.д. ознаки цієї групи дозволяють швидко розпізнавати або виділяти із сукупності цілком визначені елементи. Ознаки особливостей конструктивного виконання мають допоміжне значення і можуть бути замінені ознаками інших груп. Тому цю групу ознак рекомендовано використовувати при неможливості або завеликій складності опису ТР ознаками інших, власне, груп.

5. Геометрична форма елементів. Любий елемент ТС має певну геометричну форму або приймає форму (рідини, газу) взаємопов'язаного з ним елемента. Кількість різних елементарних ознак геометричної форми досить обмежена, однак в сукупності вони забезпечують отримання всього розмаїття відомих в техніці форм. Як правило, елементи, що мають складну геометричну форму, можуть бути підрозділені на елементи більш низького порядку, які мають більш просту геометричну форму.

6. Матеріал елементів. Вибір того чи іншого матеріалу дозволяє отримати необхідні властивості як окремих елементів, так і ТС в цілому. Слід зауважити, що розмаїття ознак цієї групи швидко збільшується у зв'язку із створенням і отриманням нових матеріалів із різними властивостями.

7. Співвідношення параметрів (розмірів, кількості елементів, значення мас, об'ємів і т. д.). Ця група ознак використовується порівняно рідко. Співвідношення параметрів може бути виражено у числовій формі або у вигляді певної математичної залежності.

Наведені групи ознак дозволяють описувати ТР з достатнім ступенем повноти. В залежності від складності ТР і ступеня деталізації при його описі використовуються різні групи ознак. Так, при розбитті ТС на елементи більш високого рівня (блоки, вузли і т.д.)

Лекція 9. Технічне забезпечення процесу проектування використовуються, в основному, групи ознак 1-3, а при більш детальному розбитті(деталі, елементи деталей) – групи 4-7.

Як показали дослідження, найбільш простим і універсальним засобом опису ТР за допомогою приведених груп ознак є граф типу дерева, який далі будемо називати деревом ТР. Вершини графу відповідають елементам ТР і їх ознакам, а дуги показують підпорядкованість між елементами, а також між елементами і їх ознаками. По інформаційному змісту дерево ТР будується від загального до часткового в напрямку від кореня до вершин. Тому більшість ознак елементів знаходяться на кінцевих вершинах дерева. Наприклад, ТР, представлене у вигляді дерева, показане на рисунку 2. Цифра перед текстом у кожній вершині відповідає номеру групи ознак. З рисунку видно, що опис ТР у вигляді ієрархічної деревовидної структури, що включає впорядкований перерахунок елементів і їх ознак, містить необхідну і достатню інформацію про ТР як про цілісну конструкцію.

Представлення окремого ТР у вигляді ієрархічного дерева виконується в наступному порядку:

- 1. Вивчають текстовий і графічний опис розглянутої ТС або її натуральний взірць; визначають її функцію, фізичний принцип дії(ФПД), склад;
- 2. Аналізують функції ТС, в результаті чого отримують ієрархічну функціональну структуру;
- 3. функціональну структуру доповнюють конструктивними ознаками ТС і її елементів. При цьому виявлення суттєвих ознак і їх зв'язків з елементами доцільно починати з ознак, що відносяться до елементів 1 рівня, потім – до елементів другого рівня і т.д. Для спрощення процесу виділення суттєвих ознак із всієї множини існуючих рекомендовано наступний порядок включення ознак у дерево: взаємозв'язок елементів, взаємне розташування елементів, геометрична форма елементів, матеріал елементів, співвідношення параметрів.

Суттєвість ознак оцінюється по ступеню їх впливу на виконання основних функцій ТС. Таким чином, після виявлення ознак і встановлення їх зв'язків з функціональними елементами отримуємо опис ТР у вигляді дерева(наприклад, дерево ТР плоского електричного фонаря на рис.2). Аналогічні дерева можна побудувати для структури любой МЕМС (див. Теслюк В.М.).

Побудова дерева ТР неможлива без встановлення єдиної термінології по обраному класу об'єктів, яка використовується для опису об'єктів ТР і їх ознак. Єдина термінологія виключає похибки, що виникають при кодуванні інформації і забезпечує єдине розуміння опису ТР. Терміни повинні, по можливості, відображати функції, що виконуються, і приналежність їх до конкретних конструктивних елементів.

В зв'язку з швидким зростанням кількості різних ТС і їх елементів можуть виникнути термінологічні проблеми, пов'язані з довільним присвоєнням елементам їх найменувань, що не мають ніякого зв'язку з функцією, що ними виконується.

систем

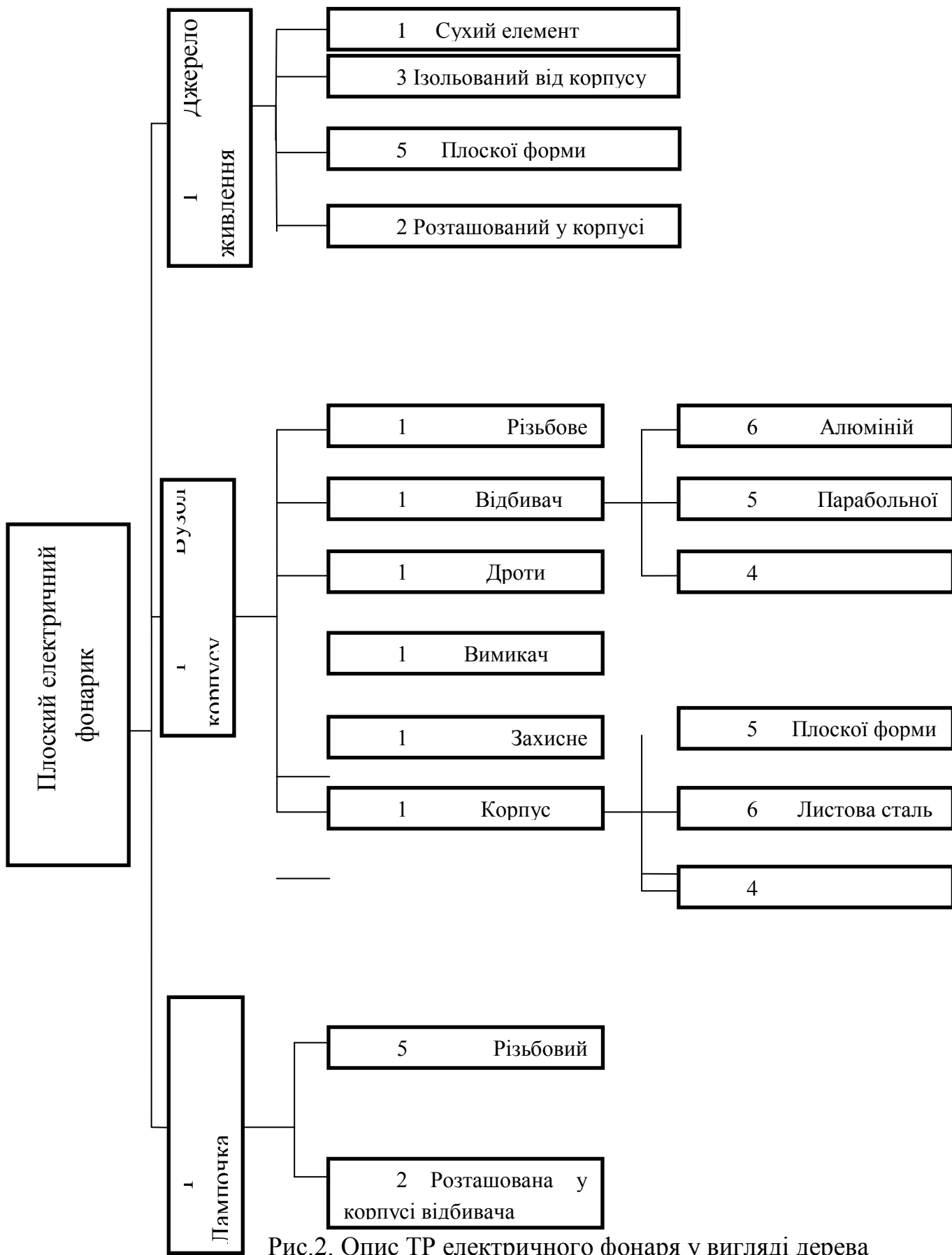


Рис.2. Опис ТР електричного фонаря у вигляді дерева

Лекція 9. Технічне забезпечення процесу проектування

Тому створювати нові технічні терміни бажано на основі єдиного функціонального підходу. Це дозволить значно скоротити кількість технічних термінів, підвищить їх загальнодоступність, розуміння і спростить автоматизовану обробку інформації, пов'язану з проектуванням і використанням ТС.

Синтез технічних рішень

Дослідження технічних завдань і технічних рішень

Перевірка повноти списку вимог в ТЗ.

В процесі розробки і експлуатації системи пошукового конструювання накопичується масив різних ТЗ у вигляді повних списків вимог. З масиву вибирають найбільш близькі до того ТЗ, що пред'явлено на експертизу. Порівняльний аналіз пред'явленого ТЗ з близькими аналогами дозволяють виявити неповноту ТЗ, і потім його уточнити. Перевірку повноти можна проводити також порівняльним аналізом відповідного ТЗ з масивом вимог.

Перевірка ТЗ на можливість реалізації.

Ця перевірка проводиться введенням ТЗ в ЕОМ для синтезу допустимих технічних рішень. При вводі ТЗ необхідно перевірити (по найменуванням і значенням вимог) відповідність ТЗ і масиву вимог до класу ТР. При неспівпадінні найменувань можлива еквівалентна заміна не співпадаючих вимог ТЗ з вимогами з масиву. Якщо така заміна не коректна (або неможлива), то або масив вимог доповнюють новою вимогою, або по новій вимозі проводять експертну оцінку отриманих допустимих ТР. Якщо по введеному ТЗ є допустимі ТР, перевірку на можливість реалізації можна закінчити. Якщо допустимих ТР немає, або виводять для наступного аналізу і доробки кращі неприпустимі ТР, або починають змінювати значення вимог і самі вимоги для отримання найбільш прийнятної ТЗ, що має допустимі ТР.

Слід зауважити, що навіть у випадку можливості реалізації ТЗ є зміст дослідити підсилення його окремих вимог або введення нових або бажаних, для чого можна розробити відповідну діалогову процедуру.

Виявлення патентоспроможних варіантів ТР.

При проведенні на попередньому кроці дослідження ТЗ з отриманням деякої множини допустимих і недопустимих ТР паралельно працюють патентознавці по розглянутому класу ТС для виділення множини ТР, що представляє інтерес з точки зору їх наступної експертизи на патентоздатність. При формуванні множини потенційно патентоздатних варіантів ТР рекомендується більш широко провести зміни списку вимог і їх значень, аналізуючи також і неприпустимі ТР. Отримані потенційно патентоспроможні ТР, ясна справа, потребують подальшої проробки в аспекті складання заявок на винаходи.

Оцінка рівня якості і перспективності ТР.

Рівень якості допустимого ТР визначається екстремальними значеннями певних його вимог (показників) порівняно із значеннями цих вимог в кращих світових взірцях. Для оцінки рівня якості досліджується ТЗ, що відповідає представленому ТР, для його підсилення і отримання покращених допустимих ТР. В результаті дослідження дається оцінка рівня якості розглянутого ТР, і у випадку знаходження покращених ТР даються рекомендації по підвищенню якості ТР, що розглядаються.

ЗАСТОСУВАННЯ МАТЕМАТИЧНОГО ПРОГРАМУВАННЯ В ПОШУКОВОМУ КОНСТРУЮВАННІ

Класифікація задач математичного програмування

Для оцінки перспективності методів МП в розв'язанні задач пошуку найбільш раціональних і нових ТР необхідно мати більш-менш повний перелік таких задач. Отримання такого переліку і власне прогнозування методів МП представляє собою досить складну задачу. В зв'язку з цим розглянемо один з можливих підходів, який хоч і не дає вичерпної відповіді, однак дозволяє отримати деяке представлення про проблему. Суть цього підходу полягає в тому, що більш широкий перелік задач МП можна отримати аналізуючи фонд евристичних прийомів, який містить узагальнений список процедур перетворення і синтезу ТР. Цей список повинен включати більшість різноманітних по формі і змісту задач пошуку нових ТР.

Слід зауважити, що кількість евристичних прийомів досить велика (кілька сот) і майже кожен з них містить інженерну генетику деякої часткової задачі. Необхідно ж сформулювати по можливості узагальнені класи і підкласи задач математичного програмування (МП), кількість яких повинна бути менша кількості часткових інженерних задач.

Якщо підійти до цієї проблеми з чисто математичної точки зору, то майже всі задачі, що розглядаються у фонді евристичних прийомів, можна поділити на 3 класи, тобто на пошук оптимальних значень змінних:

- -неперервних;
- -дискретних;
- -одночасно неперервних і дискретних, що представляють собою комбінацію перших двох задач.

Однак така класифікація занадто узагальнена і малопродуктивна в сенсі розробки ефективних методів МП. З іншої сторони, досвід розробки і застосування різноманітних методів МП показав, що найбільш ефективні методи рішення часто пов'язані з деякою узагальненою геометричною, фізичною або інженерною специфікою виділеного класу задач. Якраз на основі використання і вдалось отримати найбільш сильні методи і процедури рішення, як це, наприклад, вдалось для задач пошуку оптимальних форм, вибору оптимального розміщення і т.д.

Ця обставина і покладена в основу принципу виділення і класифікації задач МП.

Аналіз фонду евристичних прийомів з врахування вказаного принципу дозволив виділити вісім класів задач МП (табл.1).

Табл.1

Класифікація і характеристики задач МП

Клас математичної задачі	Перелік х-к змінних і (або) способів їх зміни, що характеризують інженерну суть задачі
1	2
1. Пошук оптимальних значень неперервних змінних	Основні параметри і показники виробу: габарити, об'єм, довжина, кількість речовини, енергії; час дії; зниження точності або стабільності; підсилення шкідливих

Лекція 9. Технічне забезпечення процесу проектування

	факторів, локальна концентрація сили, напруги; організація резервних можливостей.
2.Пошук оптимальної форми елементів	Скручування або згин; зміна кількості отворів і полостей, включно з їх утворенням і виключенням; порушення або утворення симетрії; перехід від плоских до ліній поверхонь і навпаки; періодична зміна форми, отримання форми з протилежними властивостями; виключення місцевих послаблень; використання форми інших об'єктів; довільна змінна форми (термінатор і трансформери)
3.Пошук оптимальної компоновки і розміщення	Дискретні зміни: горизонтальне-вертикальне-похиле положення; поворот на 45^0 , 90^0 , 180^0 ; заповнення собою вільного простору; утворення і заповнення вільного простору; розміщення в точці – по лінії – декільком лініям – в площині – в декількох площинах – в просторі; зміна напрямлення дій; функціонально вигідне або безпечне розташування робочих органів та інших елементів; довільна зміна схеми розміщення; послідовне, паралельне або змішане з'єднання; з'єднання елементів гнучким зв'язком
4.Пошук оптимальних параметрів руху	Параметри руху; напрямок руху або обертання; поступальний-обертальний рух; складна траєкторія - рух по прямій або колу; рух по лінії – декільком лініям – по площинам – декільком площинам – у трьохвимірному просторі; виключення руху важкої частини; виключення вільних/зворотніх і холостих ходів; нерухоме – рухоме фізичне поле; традиційних рух – періодичний або коливальний; зміна степені свободи переміщення; сильна зміна швидкості та інших параметрів руху; моделювання руху органів людини та інших об'єктів (дельфіни).
5.Пошук оптимальних параметрів і складу матеріалів	Параметричні властивості матеріалу; поверхневі властивості елементів; зробити матеріал взаємодіючих елементів однаковим; використати пористий, композиційний або прозорий матеріал або із змінними у часі характеристиками; використати для кожного елемента найбільш підходящий матеріал; виключити зайвий матеріал; можливі матеріали гнучкі – рідкі – газоподібні; використати дешевий, найновіший або еквівалентний матеріал; виключити шкідливі або небажані домішки; замінити матеріал поверхні тертя, повітряне середовище; використати речовини, аналогічні ферментам або каталізаторам.

систем

6. Задачі динамічної оптимізації	Основні показники і параметри; геометрична форма; розміщення і компоновка; параметри руху; параметри фізичного поля; характеристикки і властивості матеріалу; постійна змінна; періодична або імпульсна задачі енергії; асинхронна-синхронна дія; замінити параметр зворотнього зв'язку.
7. Задача стандартизації і уніфікації	<u>Виключити</u> : підбір і підгонку елементів. <u>Використати</u> комплексну уніфікацію; принцип агрегування. Зробити елементи взаємозамінними; уніфіковані або стандартні елементи для сімейства об'єктів. <u>Замінити</u> : різні спеціалізовані об'єкти одним універсальним; різні елементи одним уніфікованим; декілька однорідних елементів одним стандартним. Розділити об'єкти на уніфіковані частини; об'єднати стандартні або уніфіковані елементи.
8. Пошук оптимальних значень дискретних змінних	Кількість однорідних елементів, працюючих машин, двигунів, органів, деталей; <u>розподіл пам'яті по рівнях ієрархії</u> ; степінь дроблення об'єкта; послідовність операцій. <u>Виключити</u> : поверхні стержня; найбільш навантажений елемент; непотрібні або шкідливі проміжки часу; роз'ємність деталей; механічну обробку. <u>Використати</u> : новий елемент, агрегат, обладнання; автоматичну подачу мастил; розсувну конструкцію; автономне управління і привід; дублюючі або запасні елементи; прикриття елементів, які можна легко пошкодити або всього об'єкту; силу органів людини (3D проектування); перевантаження. <u>Замінити</u> : зжим розтягненням або стягненням; ковзання катанням; коштовні або елементи, що легко пошкоджуються, моделями, макетами, дешевими недовговічними елементами; модель об'єкта; джерело енергії; тип приводу; розміщення, механічну схему -- тепловою-пневматичною-електричною-оптичною-електронною; зв'язки і способи з'єднання елементів; послідовні операції на паралельні. <u>Поділити (розділити) на окремі частини</u> : крихкий елемент; потік, що рухається; ємність; діючу силу. <u>Об'єднати</u> органи управління і контролю; операції. Зробити елемент таким, що відокремлюється; ізолювати елемент від середовища (корпусовані мікросхеми).

Лекція 9. Технічне забезпечення процесу проектування

Зауважимо деякі особливості кожного класу задач.

- *Задачі пошуку оптимальних значень неперервних змінних* відносяться до пошуку оптимальних значень параметрів заданих ТР.
- *Задачі пошуку оптимальної форми елементів* з інженерної точки зору представляють собою один із самих обширних класів, оскільки деталі, вузли і навіть самі складні вироби мають цілком визначену форму, яка звичайно обумовлена рядом обмежень і критеріїв. В патентознавстві виділена спеціальна група ознак новизни по геометричній формі елементів; по наявній статистиці в галузі машинобудування, приладобудування і будівництва від 20 до 30% патентів видають на конструктивні особливості по формах елементів.
- *Задачі пошуку оптимальної компоновки або розміщення* з інженерної точки зору також представляє собою один із самих обширних класів, оскільки кожна деталь і вузол ТС мають, як правило, певну орієнтацію і положення в просторі по відношенню до інших деталей і вузлів. В свою чергу, кожна автономна ТС має також певне положення в просторі і по відношенню до інших ТС. В патентознавстві також виділена спеціальна група ознак новизни по взаємному розташуванню елементів.
- *Задачі пошуку оптимальних параметрів руху* досить широко розповсюджені переважно в галузі механічних пристроїв і процесів. В цьому класі розглядається задачі пошуку стаціонарних по часу значень параметрів руху. В патентознавстві особливості руху вкл., в основному, в групу ознак новизни по взаємозв'язку елементів і особливостям їх конструктивного виконання.
- *Задачі пошуку оптимальних параметрів і складу матеріалів* також відносяться до одного з найбільш обширних класів, тому що прогресивний розвиток більшості ТС пов'язаний з використанням нових матеріалів і речовин і їх комбінацією з традиційними матеріалами. В патентознавстві виділена спеціальна група ознак новизни по матеріалах, з яких виконані елементи.
- *Задачі динамічної оптимізації* або оптимального управління мають місце в ТС, де є доцільним зміна в часі різних параметрів і характеристик.
- *Задачі стандартизації та уніфікації* складають досить широкий і швидко зростаючий клас, оскільки з прискоренням технічного прогресу зростає роль стандартизації.
- *Задачі пошуку оптимальних значень дискретних змінних.* Цей клас включає досить різноманітні задачі, які, на відміну від 2-7 класів, не мають спеціальних спільних властивостей. З цього класу виділяють більш вузькі спеціальні підкласи, оскільки у нього увійшли всі решта задач, до яких можна застосувати підходи МП. (гілки-границі: розмаїття застосування)

Алгоритми пошуку глобально-оптимальних рішень

Рекомендовані алгоритми пошуку глобального екстремуму

Розглянемо алгоритми, які можна використати як для розв'язання задач параметричної оптимізації (ПО), так і для задач структурної оптимізації (СО). Укрупнена схема алгоритмів пошуку глобально-оптимальних рішень в задачах СО складається з 4 блоків або процедур (рис.3):

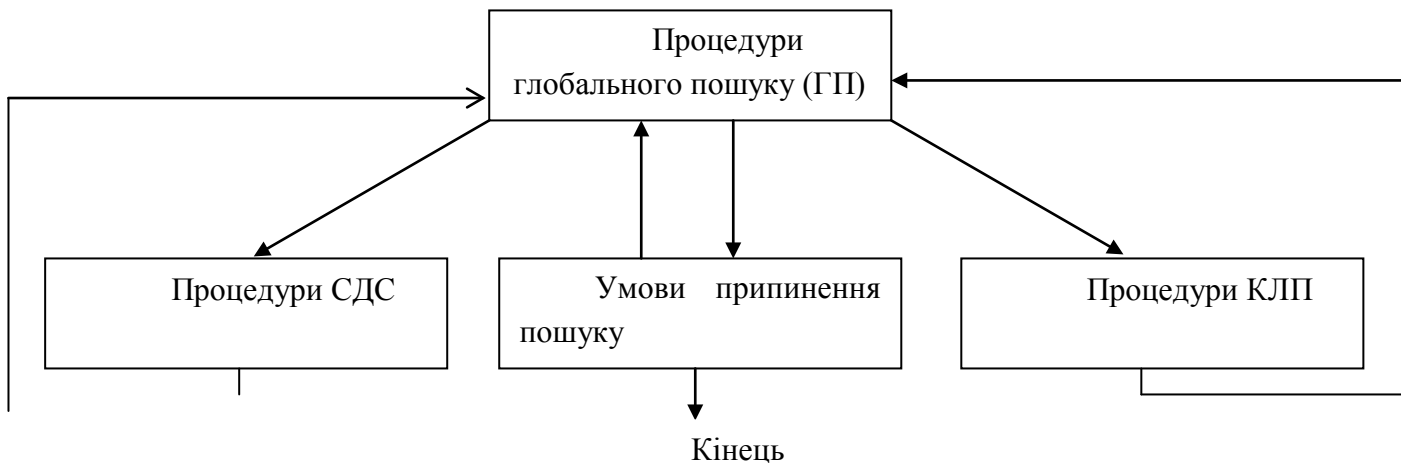


Рис. 3. Схема алгоритму пошуку глобального екстремуму

1. *Процедури синтезу допустимої структури (СДС)*, що забезпечує вибір допустимого рішення з будь-якої підобласті всієї області пошуку;
2. *Процедури кроку локального пошуку (КЛП)*, що забезпечують перехід від одного рішення до іншого допустимого рішення, як правило, тої ж структури, але з покращеним значенням критерію; під кроком локального пошуку можна розуміти деякий умовний крок по якому алгоритму пошуку локального екстремуму (напр., одна ітерація по методу найшвидшого спуску);
3. *Процедури глобального пошуку (ГП)* які керують роботою процедур СДС і КЛП;
4. *Процедура перевірки умов припинення пошуку*, яка визначає закінчення розв'язання задачі.

В деяких випадках побудову процедури СДС можна звести до попереднього складання набору допустимих структур, з якого робляти вибір при кожному звертанні до процедури СДС. Якщо суть цієї процедури полягає у виборі по можливості допустимого набору змінних СО, то буде корисним включити в неї правила вибору змінних, які побудовані на евристичних міркуваннях, аналітичних або експериментальних дослідженнях, вивчені досвіду проектування і експлуатації аналогічних ТС. Для деяких складних або маловивчених задач проектування вважаю побудувати процедуру СДС, що забезпечує отримання допустимих структур. В цьому випадку в процедуру доцільно включити операції перетворення недопустимих структур в допустимі. Набір таких операцій можна скласти з підходящих евристичних прийомів. Перетворення недопустимих структур в допустимі можна також вирішувати як задачу оптимізації. В діалоговому режимі роботи функцією процедури СДС може взяти на себе проектувальник.

В цілому по процедурі СДС можна надати наступні рекомендації, направлені на підвищення ймовірності набору допустимих структур і зниження обсягів обчислень по оцінці недопустимих:

1. Способи вибору значень змінних повинні мати правила, які відсікають завідомо нераціональні і недопустимі значення змінних і їх комбінацій;
2. Перевіряти обмеження необхідно не після побудови структури в цілому, а по можливості в

Лекція 9. Технічне забезпечення процесу проектування процесі побудови, що дозволяє скоротити зайву роботу по непотрібних побудовах і у ряді випадків одразу внести правки по усуненню дефектів структури;

3. Обмеження, що перевіряються, повинні бути впорядковані по зниженню ймовірності їх порушення, таке впорядкування іноді можна проводити автоматично в процесі розв'язання задачі.

Процедури КЛП включають звичайно способи зміни змінних, орієнтовані на розв'язання задач як СО, так і ПО. Рекомендації по побудові процедур СДС можна використати і при побудові способів локальної зміни дискретних змінних. Для зміни неперервних змінних, як правило, використовують різні алгоритми локального пошуку. Розглянемо ті, що переважають.

Розглянемо алгоритми пошуку глобального екстремуму, які рекомендовано у якості процедур. Ці алгоритми будемо відповідно називати алгоритмом усічення і конкуруючих точок.

В основу запропонованих алгоритмів покладена ідея конкуренції рішень, яка реалізує наступні принципи:

- Пошук глобального екстремуму здійснюється декількома конкуруючими рішеннями (точками);
- Умови конкуренції однакові для всіх рішень;
- В певні моменти деякі «гірші» рішення бракують;
- Послідовний локальний спуск кожного рішення (спочатку грубий, а потім все більш точний) проходить незалежно від спуску інших рішень.

Конкуренція дозволяє за рахунок відсіювання рішень, що спускаються в локальні екстремуми, достатньо швидко знаходити глобальний екстремум в задачах, для яких значення функціонала, усередненого по всій області пошуку, а басейн глобального екстремуму не дуже малий.

Для зручності опису алгоритмів будемо рішення також називати точкою (в багатовимірному просторі пошуку) і незалежно від того, чи розв'язується задача ПО, чи СО, позначати його через X .

Алгоритм усічення (Меркур'єв В.В.) в загальному вигляді включає наступну послідовність операцій:

1. Маємо m_i допустимих початкових точок

$$x_1^i, \dots, x_{m_i}^i \quad (6)$$

Початкові ($i=0$) точки отримують по деякій процедурі СДС. i означає, що точки (6) зробили i кроків локального пошуку. Причому КЛП залежить від i , відповідно, для роботи алгоритму необхідна процедура КЛП.

2. Серед m_i точок (6) відбираємо m_{i+1} точок, найкращих по значеннях мінімізуємої функції (або по іншому критерію).

$$\tilde{x}_1^{i+1}, \dots, \quad (7)$$

Нехай при цьому φ_i - значення функції в найгіршій з вибраних точок, тобто

$$\varphi_i = \max f(\tilde{x}_j^{i+1}) \\ j = \overline{1, m_{i+1}}$$

3. Кожній відібраній точці дозволяється зробити ($i+1$)-й КЛП в результаті чого точки (7) перейдуть в

$$x_1^{i+1}, \dots, x_{m_i+1}^{i+1} \quad (8)$$

П.п. 1-3 повторюються, доки i не буде дорівнювати T . При цьому здійснене в п.1-3 (при збільшенні індексу i на одиницю) будемо називати глобальним кроком. В п.2 в якості критерію відбору точок можна використати критерії, відмінні від мінімізуємої функції. Наприклад, для деяких задач доцільно використати

$$f(\vec{X}) + \alpha \frac{\partial f(\vec{x})}{\partial l}$$

Де α – деяка додатня величина;

$\frac{\partial f(\vec{x})}{\partial l}$ – приріст функції в напрямку кроку локального пошуку.

Методи рішення задач багатокритеріальної оптимізації

Підходи до рішення задач оптимізації при наявності декількох критеріїв

При проектуванні складних ТС, наприклад, літака, ЕОМ, станка, досить часто важко вибрати і формалізувати єдиний критерій оптимальності і в результаті звести задачу конструювання до задачі математичного програмування (МП) виду:

$$\phi(\vec{\delta}) \rightarrow \min, \vec{\delta} \in D \quad (9)$$

В таких випадках конструктор змушений звертатись до лиця, що приймає рішення (ЛПР), яке, зазвичай, має більше інформації про мету проектування.

Іноді можна розраховувати на те, що у ЛПР існує хоча б неформальний критерій який забезпечує вибір найкращого у будь-якій парі рішень. При цьому формально задача проектування зводиться до задачі (9), в якій за розрахунком значень критерію $\phi(\vec{\delta})$ приходиться звертатись до ЛПР. Однак занадто часті звертання до ЛПР, необхідні для розв'язування задачі (9), здійснити практично не реально.

Вказані труднощі можна значно зменшити, якщо вдасться виділити набір **локальних критеріїв** $f_i(\vec{\delta})$, які є порівняно простими функціями від вектору оптимальних параметрів, а критерій $\phi(\vec{\delta})$ - монотонна функція локальних критеріїв

$$\phi(\vec{\delta}) = \psi(F(\vec{\delta})) = \psi(f_1(\vec{\delta}), f_2(\vec{\delta}), \dots, f_m(\vec{\delta})) \quad (10)$$

(Монотонність означає, що з $f_i(\vec{\delta}') \geq f_i(\vec{\delta})$ для всіх $i = \overline{1, m}$, випливає $\phi(\vec{\delta}') \geq \phi(\vec{\delta})$)

Без втрати узагальнення будемо вважати, що покращенню ТС відповідає мінімізація всіх локальних критеріїв. Якщо якісь локальні критерії по постановці задачі вимагається максимізувати, то їх приводять до мінімізованої форми (беруть із зворотнім знаком, або у зворотній формі $\frac{1}{f_i(\vec{\delta})}$ відповідно). Дані $\phi(\vec{\delta})$ будемо називати **глобальним критерієм**.

Локальними критеріями можуть бути часткові характеристики ТС, такі як: маса, габарити, вартість, надійність; для літака – крейсерська швидкість, потолок, дальність польоту, витрати

Лекція 9. Технічне забезпечення процесу проектування

пального; для ширококутового підсилювача – коефіцієнт підсилення, нерівномірність коофіцієнт в полосі, рівень власних шумів, кількість нелінійних спотворень.

Слід зауважити, що навіть у випадку успішної формалізації критерію $\phi(\vec{\delta})$ постановки задачі (9) часто виникають непереборні проблеми із-завеликої кількості оптимальних параметрів, занадто великих втрат часу на визначення значень критерію $\phi(\vec{\delta})$, який, як правило, досить складно залежить від $\vec{\delta}$. У таких випадках також є більш доцільним зведення задачі (9) до задачі багатокритеріальної оптимізації (У практиці проектування часто зустрічаються задачі багатокритеріальної оптимізації у своїй природній постановці. Наприклад, необхідно сконструювати пристрій, що має мінімальну масу, максимальну надійність і мінімальне тепловиділення).

При отриманні (10) замість (9) можна розв'язати задачу пошуку оптимального рішення на множині Парето Π

$$\phi(\vec{\delta}) \rightarrow \min, \vec{\delta} \in \Pi \quad (11)$$

$$\Pi = \{ \vec{\delta} \in D : \exists x^{\circ} \in D : f_i(\vec{\delta}^{\circ}) \leq f_i(\vec{\delta}), i = \overline{1, m}; F(\vec{\delta}^{\circ}) \neq F(\vec{\delta}) \}, \quad \text{або}$$

початкову задачу

$$\psi(\vec{F}) \rightarrow \min, \vec{F} \in \Pi_F, \quad (12)$$

де $\Pi_F = F(\Pi)$ - образ множини Парето в просторі локальних критеріїв.

Цей прийом(неформальна декомпозиція) складається з 2-ох етапів: 1) виділення множини Парето Π або Π_F ; 2) рішення (11) або (12). Перший етап не залежить від глобального критерію $\phi(\vec{\delta})$. Його реалізація дозволяє значно скоротити необхідну для визначення оптимального рішення кількість розв'язків $\phi(\vec{\delta})$. Декомпозиція, як правило,

носить неформальний характер, так як вибір **векторного критерію** \vec{F} , який правильно відображає змістовну сторону критерію $\phi(\vec{\delta})$, є для конструктора складною задачею із-за його неповної інформованості. Проблему вибору часткових показників необхідно розв'язувати разом з ЛПР на етапі формування і конкретизації мети проектування.

Далеко не завжди вдається добитися остаточної ясності про мету проектування. Якщо набір локальних критеріїв вибраний правильно, тобто характеризує всі найбільш суттєві властивості ТС, що цікавлять ЛПР, то набір реалізованих альтернатив, який завідомо має оптимальний розв'язок, представляє множину Парето. При цьому векторний критерій і отримана на його основі множина Парето грають більш важливу роль і окрім декомпозиції дозволяють організувати діалог конструктора(оператора) з ЛПР для конкретизації мети проектування.

Таким чином, найпростіший шлях рішення полягає у виділенні множини Парето і організації на ній пошуку оптимального рішення. В дискретному випадку(хоча б в принципі) можна виділити всю множину Парето, а в неперервному – побудувати скінченну точкову множину Π_{ε} множини Парето, що покриває його із заданою похибкою ε . Однак в багатьох випадках виділення Π_{ε} представляє величезну по обсягу обчислень задачу. Це обумовлено тим, що необхідно будувати ε -сітку для множини розмірності $m-1$, а для отримання кожної точки сітки розв'язати задачу МП того ж порядку складності в сенсі рельєфу мінімізуємої

систем

функції, що і (9):

$$\varphi(\vec{F}(\vec{\delta})) \rightarrow \min, \vec{\delta} \in D \quad (13)$$

де $\varphi(\vec{F})$ - згортка векторного критерію. Задача (13) відрізняється від (9) тим, що φ звичайно дуже проста функція від \vec{F} . При більш або менш значній кількості локальних критеріїв (5÷10) і при високих вимогах до точності покриття (мале ε) кількість точок ε -сітки і, відповідно, кількість розв'язуваних задач МП виду (13) стають занадто великими.

Деяке полегшення буде при одночасному пошуку набору паретовських точок, коли, розумно організувавши обчислення, можна суттєво зменшити їх об'єм порівняно з багаторазовим розв'язанням задачі (13). Така задача формулюється у вигляді:

$$\varphi_i(\vec{F}(\vec{\delta})) \rightarrow \min, \vec{\delta} \in D, i = \overline{1, l}. \quad (14)$$

де l - кількість точок в наборі, а φ_i - згортка векторного критерію. Специфіка (14) полягає в тому, що φ_i - дуже прості функції від \vec{F} і обчислення їх значень в точці $\vec{\delta}$ по вище визначеному значенню векторного критерію в цій точці значно простіше, ніж визначення значення $\vec{F}(\vec{\delta})$. Тому знайшовши один раз значення $\vec{F}(\vec{\delta})$ в точці x без особливих додаткових витрат можна визначити значення всіх згорток в цій точці.

Якщо для отримання паретовських точок використовується метод перебору по деякій сітці на D , то за один прохід (значення векторного критерію в кожній точці визначається тільки один раз) можна знайти всі l паретовських точок. Цей підхід можна рекомендувати для невеликої кількості оптимальних параметрів (не більше 10).

3.4. Алгоритми пошуку оптимальних форм

Рекомендовані алгоритми

Враховуючи специфіку задач оптимізації форм (велика розмірність, деяка зв'язність параметрів, не лінійність і (або) не диференційованість цільових функцій, велика кількість різного роду обмежень), використання детермінованих методів пошуку для розв'язання таких задач або неефективне, або взагалі неможливе. Тут під ефективністю пошуку розуміють витрати машинного часу на пошук і точність знайденого рішення за заданий проміжок часу. В цих умовах найбільш прийнятними є статистичні методи пошуку, в тому числі спеціальні алгоритми випадкового пошуку.

Алгоритми, що рекомендовані, можна поділити на дві групи: В алгоритмах першої групи реалізовано принцип «покарання випадковістю», тобто випадковий крок виконується після попереднього невдалого, а у випадку вдалого кроки повторюються в одному і тому ж напрямку. Таким чином в цих алгоритмах застосовується лінійна тактика спуску. Алгоритми другої групи побудовані за принципом «заохочення» випадковістю, тобто після кожного вдалого кроку вибирають новий випадковий крок. В цих алгоритмах застосовується нелінійна тактика пошуку. Аналогами статистичних алгоритмів першої і другої групи є детерміновані алгоритми: відповідно найшвидшого спуску і градієнтний. Основна відмінність алгоритмів випадкового пошуку від детермінованих аналогів – спосіб вибору напрямку кроку. В

Лекція 9. Технічне забезпечення процесу проектування
рекомендованих використовується два способи вибору напрямку: чисто випадковий і парними пробами(роблять два кроки з однієї точки в протилежних напрямках, і за напрямком робочого кроку вибирають напрямок кроку з меншим значенням цільової функції).

Принципова відмінність запропонованих алгоритмів від відомих алгоритмів випадкового пошуку полягає в способі вибору приростів($\Delta\vec{d}$). На кожному кроці пошуку в запропонованих алгоритмах прирости отримують не всі змінні, а групи змінних, тобто **пошук ведуть не у всьому просторі параметрів, а в підпросторах**. Розмірність підпростору при кожному виборі кроку пошуку визначається випадково (від 1 до n). Інша відмінна особливість цих алгоритмів полягає у наявності **функціонального зв'язку між приростами змінних**, що входять в склад підпростору. Наявність такого зв'язку забезпечує складність зміни розмірів, що оптимізуються в ході пошуку, таким чином враховується, зв'язність параметрів в задачах оптимізації форм.

12. Евристичні методи пошукового конструювання нових технічних рішень

Класифікація евристичних методів

Відомі різні методичні засоби розв'язання задач пошукового конструювання: принципи, правила, прийоми, методи, методики, алгоритми і т.д. Під евристичним методом розуміють послідовність процесів або процедур обробки інформації, яка виконується з метою пошуку більш раціональних і нових конструктивних рішень. Для такої послідовності можуть бути відсутні доведення, що вона є найкращою в сенсі швидкодії або трудомісткості розв'язання задачі, і немає варіантів знаходження найкращого або глобально-оптимального рішення. Як правило, евристичні методи в певній ситуації доцільно використовувати, оскільки в цих випадках інші математично обґрунтовані методи є менш ефективними або взагалі відсутні.

З точки зору принципової можливості використання ЕОМ в розв'язанні задач пошукового конструювання можна виділити три групи методів.

- 1. Повністю формалізовані – алгоритми, які можуть бути реалізовані у вигляді програми, яка виконується в автоматичному режимі. В такій програмі чітко виділяють вхідні дані, що задаються людиною, і отримані за допомогою ЕОМ результати. Слід зауважити, що до результатів тут відносять і відсутність рішення або отримання неприпустимих рішень, які не задовольняють розробника.
- 2. Частково формалізовані евристичні методи – евроритми, в яких частина процедур обробки інформації може бути описана у вигляді алгоритмів і відповідно запрограмована на ЕОМ, а інша – евристики – реалізується людиною-оператором в інтерактивному режимі в процесі діалогу.
- 3. Повністю неформалізовані евристичні методи, що складаються тільки з набору евристик і які можуть бути реалізовані виключно засобами інтерактивного проектування.

Метод евристичних прийомів

Евристичний прийом представляє собою процес або вказівку, як перетворити існуюче або аналогічне технічне рішення(ТР), або в якому напрямку шукати, щоб отримати бажане рішення. Евристичні прийоми мають характерні для евристичних методів властивості. Більшість евристичних прийомів складається з двох частин: перша – опис простору змінних і відповідає на запитання «що змінювати» в розглянутому ТР; друга – опис способу зміни змінних і відповідає на питання «як змінювати».

Оскільки в багатьох прийомах фігурує декілька просторів змінних або(і) декілька способів зміни змінних, то такі прийоми, можна сказати, складаються з декількох пошукових процедур. Кожна пошукова процедура містить вказівку тільки на один простір змінних x , один спосіб їх зміни або один напрямок пошуку і представляє собою як би елементарний евристичний прийом. Якщо, наприклад, евристичний прийом містить два простори змінних («об'єкт», «елемент») і п'ять способів зміни змінних, то він включає 10 пошукових

Лекція 9. Технічне забезпечення процесу проектування процедур. Порівняно із безсистемним методом «проб і помилок», використання наборів евристичних прийомів може забезпечити суттєво більшу ймовірність знаходження вдалих рішень. Відомі сотні евристичних прийомів і відповідних їм пошукових процедур, які можна згрупувати, наприклад, в такі 15 груп:

1. Кількісні зміни.
2. Перетворення форми об'єктів або їх елементів.
3. Перетворення в просторі.
4. Перетворення в часі.
5. Перетворення руху.
6. Перетворення матеріалу.
7. Перетворення виключенням.
8. Перетворення добавлянням.
9. Перетворення зміною.
10. Диференціація.
11. Інтеграція.
12. Використання профілактичних мікроелементів.
13. Використання резервів.
14. Перетворення по аналогії.
15. Комбінування і комплексний синтез.

Рекомендації по застосуванню метода

При розв'язанні задач методом евристичних прийомів послідовно виконуються наступні етапи пошуку і обробки інформації:

1. З'ясування або формулювання технічного завдання – списку вимог до необхідного ТР;
2. Вибір з фонду аналогів одного або декількох прототипів, що в найбільшій мірі задовольняють ТЗ;
3. Аналіз прототипів, з'ясування їх недоліків і формулювання постановки задачі у вигляді відповідей на запитання:
 - які показники в прототипі і наскільки їх бажано покращити;
 - які нові властивості повинен мати створюваний виріб або які властивості повинен втратити розглянутий прототип;
4. Задачу починають розв'язувати з вибору найбільш підходящих прийомів з фонду евристичних прийомів. За допомогою цих прийомів перетворюють вибрані на другому етапі прототипи і аналізують можливості розв'язання поставленої задачі. При цьому слід мати на увазі, що задачу можна розв'язати не одразу, за допомогою одного прийому, а послідовно покращуючи результати за допомогою різних окремих прийомів. Іноді вдале рішення можна отримати при одночасному комплексному застосуванні двох і більше прийомів.

Якщо вибрані прийоми не дають бажаного результату, то робиться спроба застосувати підряд всі прийоми фонду. При невдалих спробах розв'язання задачі необхідно повернутись до етапу два і змінити склад прототипів або до етапу один і змінити список вимог.

Рекомендації по програмуванню і використанню методу.

Використання ЕОМ дозволяє суттєво підвищити ефективність методу евристичних

систем

прийомів. Можна рекомендувати наступні шляхи використання обчислювальної техніки.

На першому етапі можна розробити звичайну фактографічну інформаційно-пошукову систему (ІПС), що дозволяє по певних запитах вибирати необхідну інформацію (з фонду евристичних прийомів і прикладів рішення аналогічних задач, фонду матеріалів і конструктивних елементів), що значно скорочує час пошуку необхідних прийомів і даних. При цьому перетворення аналогів (синтез нових ТР і їх оцінка) залишається за людиною.

На другому етапі проводять роботи по програмуванню евристичних прийомів, що дозволяє, з одної сторони, частину роботи по синтезу передати ЕОМ, а з іншої – мати можливість створення і програмування математичних моделей по оцінці ТР, отриманих автоматичним синтезом. В деяких випадках можна створювати універсальні математичні моделі, які дозволяють оцінювати досить широкий клас рішень по деяких показниках. Це дозволяє в значній мірі автоматизувати оцінку ТР, які синтезовані конструктором або машиною.

Програми методу евристичних прийомів можна включити в ППП підсистем пошукового конструювання. Метод також рекомендовано використовувати при підготовці інформаційного забезпечення в комп'ютерних методах синтезу ТР з метою розширення множини ТР.

Узагальнений евристичний метод

Проведений аналіз різних евристичних методів пошуку нових технічних рішень дозволив встановити, що:

- кожний метод можна розділити на певну кількість послідовних етапів, призначених для розв'язування окремих під задач по пошуку і обробці інформації, а кожний етап містить деякий набір процедур для обробки інформації;
- в багатьох методах присутні однакові або близькі по змісту етапи і процедури;
- в більшості методів існує деякий інваріаційний порядок слідування етапів і процедур;
- в ряді методів використовуються інформаційні фонди, що полегшують пошук необхідних даних і отримання нових проектних рішень.

Перераховані властивості дають підставу для узагальнення евристичних методів і створення узагальненого евристичного методу. Цікавими є наступні аспекти його використання.

1. В першу чергу його доцільно використовувати як методичний посібник для розробки спеціалізованих евристичних методів, орієнтованих на певний клас технічних систем. Спеціалізований метод (у порівнянні з узагальненим) стає простіше і ефективніше за рахунок виключення ряду процедур і використання більш вузько орієнтованих інформаційних фондів.

2. Узагальнений евристичний метод і отримані з його допомогою спеціалізовані евристичні методи представляють собою добру евристичну і алгоритмічну основу для розробки інтерактивних програм пошуку нових, більш раціональних, технічних рішень. Ці програми можна використовувати як самостійні евристичні ритми, а також включати в підсистеми пошукового конструювання.

Лекція 9. Технічне забезпечення процесу проектування

3. За допомогою узагальненого методу можна покращувати відомі евристичні методи пошуку нових технічних рішень, а також отримувати модифікації відомих евристичних методів. При цьому похідна модифікація буде ефективнішою від свого оригінала. Це пояснюється тим, що в узагальненому методі більшість процедур та інформаційних масивів доповнено рекомендаціями з інших методів або спеціальними розробками, що забезпечує і підсилює отримані модифікації відомих методів.

4. Узагальнений метод можна також безпосередньо використати для рішення задач пошукового конструювання. Особливо це має сенс робити при розв'язанні окремих задач, коли недоцільною є попередня розробка спеціалізованого методу.

5. Узагальнений метод може служити методичним посібником по підготовці і перепідготовці спеціалістів. При цьому його бажано доповнювати прикладами розв'язання задач.

Структура і процедури методу

Пошук нових, більш ефективних технічних рішень, - це, перш за все, процес підготовки і обробки інформації, за допомогою якої синтезують нову інформацію у вигляді конструктивних рішень виробів або технологічних процесів. У зв'язку з цим узагальнений евристичний метод представляє собою як би опис такого процесу, умовно поділеного на сім етапів, які суттєво відрізняються цільовим призначенням. Кожний етап, в свою чергу, складається з декількох процедур підготовки і обробки інформації.

Значною і досить важливою складовою частиною узагальненого методу є наступні інформаційні фонди:

- M1 – фонд фізичних ефектів;
- M2 – фонд технічних рішень розглянутого класу технічних засобів, який включає окремий масив M2A описів технічних засобів на рівні кращих світових досягнень;
- M3 – список вимог, які висуваються до класу технічних рішень, що розглядаються;
- M4 – фонд матеріалів і конструктивних елементів, перспективних для створення нових технічних рішень;
- M5 – фонд технологічних процесів, які можна використати при виготовленні розглянутого класу технічних засобів;
- M6 – фонд евристичних прийомів;
- M7 – фонд технологічних рішень провідного класу технічних засобів;
- M8 – методи оцінки і вибору варіантів технічних рішень.

Фонди M1, M6, M8 в значній мірі універсальні і мало залежать від розглянутого класу технічних засобів. Решта фондів більш спеціалізовані і залежать від класу технічних засобів. На рис.1 приведена схема узагальненого евристичного методу, де вказані назви етапів та інформаційні фонди.

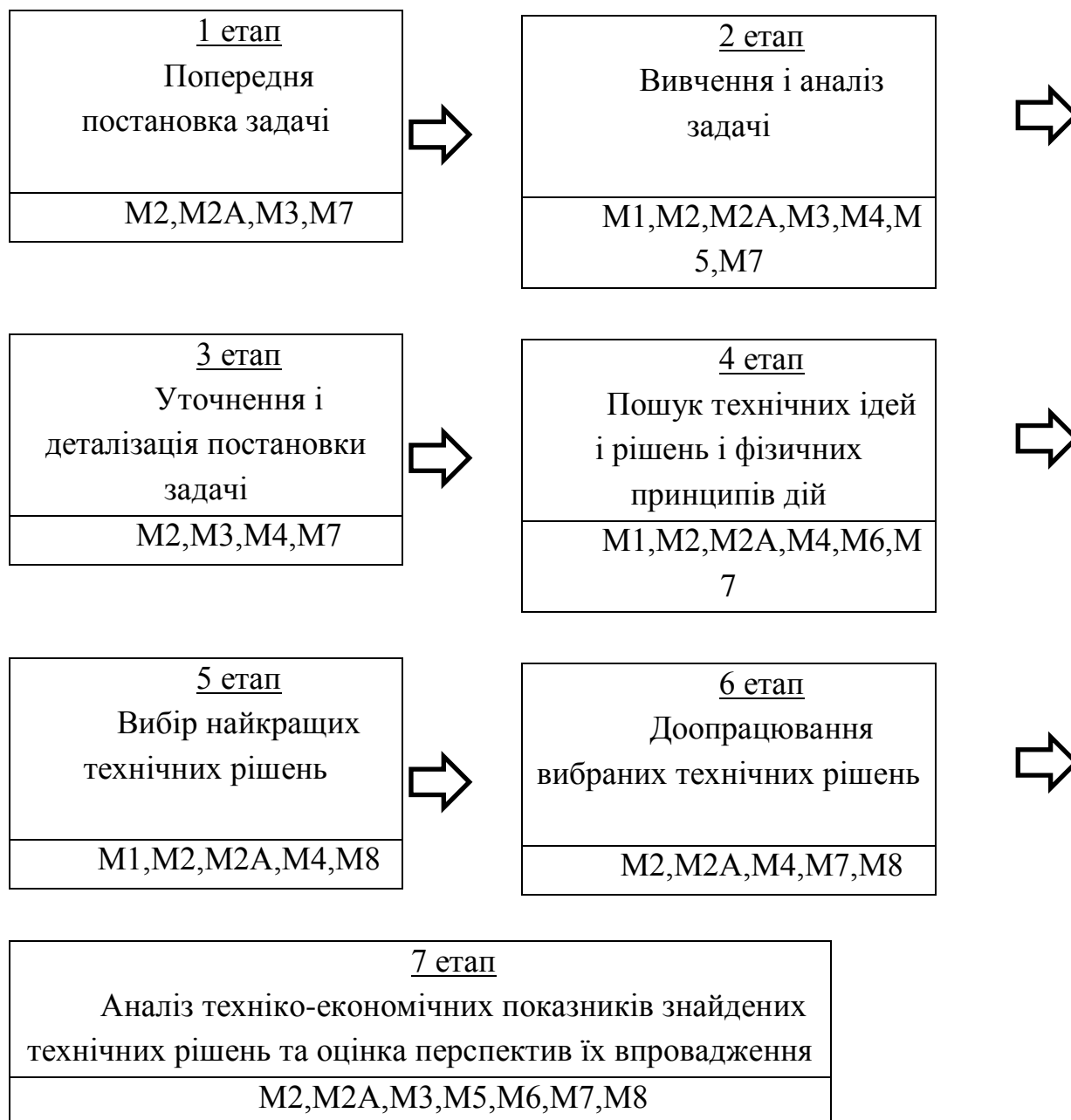


Рис.1. Схема узагальненого евристичного методу

Результати обробки інформації по рекомендаціях кожної процедури повинні відповідним чином документуватися у вигляді текстового опису, графічної схеми і т.д. Задokumentована інформація може багатократно використовуватись при виконанні інших процедур. Розглянемо детальніше ці три етапи.

1. Попередня постановка задачі. Призначення етапу – якісне і кількісне формулювання функції технічної системи, що розробляється; співставлення цієї функції з можливостями існуючих, близьких по функціям, технічних систем і виявлення незадовільних компонент функції; останні представляють собою список недоліків функціонально близьких технічних систем або список цілей рішення задачі. Формулювання функції відображає необхідність

Лекція 9. Технічне забезпечення процесу проектування створення нової технічної системи.

2. Вивчення і аналіз задачі. Призначення етапу – вивчення історії конструктивної еволюції і тенденцій розвитку розглянутого і функціонально близьких класів технічних систем для виявлення можливостей підсилення окремих показників функції; проводиться ранжування(і комбінування) списку недоліків існуючих технічних систем і виявляють причини виникнення недоліків і оцінюють умови і актуальність рішення задачі; поставлена задача розглядається як частина над системи з метою пошуку інших постановок задачі.

3. Уточнення і деталізація постановки задачі. Формулювання задачі, отримані на попередніх етапах, включають найчастіше тільки основні функціональні вимоги до технічної системи, що розробляється. Ці вимоги не враховують ряд факторів, що забезпечують працездатність і ефективність технічної системи. Тому на цьому етапі складають по можливості повний список вимог, який представляє собою технічне завдання(ТЗ) на розробку нової технічної системи. Цей список включає експлуатаційні, економічні, ергономічні та інші групи вимог з їх кількісною оцінкою. Крім того, на цьому етапі вивчають взаємозв'язки між окремими вимогами.

4. Пошук технічних ідей, рішень і фізичних принципів дії. Мета етапу – синтез розширеної множини нових технічних рішень і фізичних принципів дій, з яких потім необхідно вибрати найкраще рішення.

5. Вибір найкращих технічних рішень. Призначення етапу полягає в аналізі й оцінці всіх знайдених технічних рішень(в тому числі отриманих на основі нових фізичних принципів дій) для вибору невеликої кількості найкращих варіантів. Виконання процедур цього етапу супроводжується конструктивною розробкою, уточненням графічних ескізів і схем, кількісними оцінками і математичним моделюванням розглянутих технічних рішень або експериментальними перевірками.

6. Доробка вибраних технічних рішень. Мета етапу – більш детальна проробка вибраних на попередньому етапі технічних рішень, їх подальше покращення, експериментальна або дослідна перевірка.

7. Аналіз техніко-економічних показників знайдених технічних рішень і оцінка перспектив їх впровадження. Тут основним є наступне:

- оцінити очікуваний ефект від використання отриманих технічних рішень;
- визначити область практичного застосування отриманих технічних рішень;
- оцінити очікуваний економічний ефект в залежності від об'ємів реалізації нових технічних систем.

Рекомендації по розробці спеціалізованих евристичних методів

Склад інформації процесів узагальненого евристичного методу, широкий набір універсальних і спеціалізованих процедур пошуку нових технічних рішень дозволяють розробляти на основі узагальненого методу спеціалізовані евристичні методи, орієнтовані на пошук нових технічних рішень в певному класі технічних систем. Такий клас може формуватися в результаті виділення функціонально близьких технічних систем, що відносяться до різних галузей техніки, або в нього може входити досить вузький клас пристроїв, які цікавлять одне підприємство.

Розробці спеціалізованого методу повинне передувати вивчення специфіки необхідного

систем

класу технічних систем, тенденції їх розвитку і взаємозв'язків з іншими технічними елементами. Основний принцип побудови спеціалізованих евристичних методів полягає у виборі з узагальненого методу найбільш ефективного набору етапів і процедур обробки інформації, а також складу інформаційних фондів з їх наступною доробкою або розробкою.

Вибір етапів при розробці спеціалізованого методу для рішення конкретних класів задач не викликає, як правило, якихось ускладнень. Як правило в нього включають всі етапи узагальненого методу. Так як за допомогою спеціалізованого методу буде розв'язуватись широке коло конструкторсько-технічних задач то виключати якийсь етап без втрати якості рішення неможливо. Виключати деякі етапи можна в процесі рішення самої задачі, коли її характер, об'єм інформації про неї і вимоги до технічного рішення дозволяють більш обгрунтовано вибрати загальну схему рішення(послідовність і кількість етапів і процедур).

В цьому плані виникають проблеми при виборі складу процедур. Для скорочення трудомісткості рішення задачі бажано скоротити кількість процедур, однак, зайва мінімізація кількості процедур від'ємно впливає на якість отриманих результатів і в цілому на ефективність застосування спеціалізованого методу. Для спрощення задачі вибору всі процедури узагальненого методу поділяють на дві групи:

- інваріантні процедури, які рекомендовано включати в любий спеціалізований метод;
- додаткові процедури, які вибираються з решти процедур з врахуванням специфіки розв'язуваного класу задач і на основі експериментальної перевірки або експертної оцінки ефективності їх застосування.

Спеціалізований евристичний метод можна представити як об'єднання множини інваріантних A_i і додаткових A_g процедур обробки інформації

$$A_c = A_i \vee A_g$$

Вибір масивів інформації для спеціалізованого методу і визначення їх змісту побудовано на вивченні класу розв'язуваних задач і експертній оцінці ефективності інформаційних фондів. При цьому в першу чергу рекомендовано використовувати фонди М1, М2, М2А, М4, М6.

В процесі розв'язання задачі за допомогою спеціалізованого методу рекомендовано керуватись наступними методичними правилами:

1. Слід повністю використовувати приписи процедури, по можливості не залишаючи нез'ясованих питань. Не потрібно економити час на процедурах постановки задач, а також на документуванні результатів, отриманих після кожної процедури. При цьому бажано фіксувати всі ідеї рішення задачі; особливу увагу слід приділити графічній проробці ідей і технічних рішень. По основних функціональних елементах доцільно будувати морфологічну таблицю з використанням фонду евристичних прийомів для формування альтернатив.

2. Для скорочення часу і трудомісткості рішення наступних задач рекомендовано на основі рішення попередніх задач складати додатковий методичний матеріал, в якому систематизуються і узагальнюються результати обробки інформації по окремих процедурах.

3. Періодично рекомендовано проводити ревізію спеціалізованого методу: розвивати і доповнювати інформаційні фонди, включаючи нові ефективні процедури обробки інформації із узагальненого евристичного методу і з інших джерел, в тому числі сформульовані самими

Лекція 9. Технічне забезпечення процесу проектування користувачами, одночасно виключати малоефективні процедури і масиви інформації або їх складові частини.

Література

1. Webster Online Dictionary – wow.com
2. В. Теслюк. Моделі та інформаційні технології синтезу мікроелектромеханічних систем. – Львів: «Вежа і Ко». – 2008, 192 с.
3. А.А. Тимченко. Основи системного проектування та системного аналізу складних об'єктів.: В 2кн. – К.: «Либідь». – 2000.
4. В.М. Теслюк, П.Ю. Денисюк. Автоматизація проектування мікроелектроелектромеханічних систем на компонентному рівні. – Львів: «Львівська політехніка», 2011, 192 с.
5. Разевиг В.Д. система проектирования. OrCAD 9.2 – «Солон - Р», 2003, 528 с.
6. Л.К. Гліненко, А.А. Смердов. Технологія інженерного проектування. Структурний синтез технічних та біотехнічних систем: Навч. посібник. – Львів: Видавництво НУ «Львівська політехніка», 2004. – 388 с.
7. А.В. Жураховецький, А.Я. Яценко, Н.Б. Д'яченко. Основи технічної творчості та наукових досліджень: Навч. Посібник. – Львів: Видавництво НУ «Львівська політехніка», 2012. – 380 с.

Частина III.

Математичні моделі в проектуванні СОС

13. Роль комп'ютерного моделювання та прототипування

стають не ефективними вже сьогодні, примусили виробників самих різних галузей шукати інноваційні підходи, які дозволили б скорочувати терміни і витрати на розробку, збільшуючи тим самим конкурентоздатність як окремого виробу, так і підприємства загалом. Найбільш привабливим і оптимальним виявилось застосування комп'ютерних технологій моделювання, візуалізації й імітування, що вже давно застосовують у військово-промисловому комплексі.

Комп'ютерне моделювання дозволяє не тільки створити, але і удосконалити складний виріб, оцінити і випробувати його не на реальному підприємстві, а в середовищі віртуальної реальності. Це особливо актуально для складних, унікальних й дорогих технологічних і військово-технічних комплексів.

В сучасних САПР все більш масово застосовують технології віртуального прототипування, тобто створення віртуальної (електронної) моделі об'єкта, призначеного для подальшого виробництва. Далі такий віртуальний прототип всебічно оцінюють (напр., властивості ергономіки, безпеки, функціональності, технологічності і т.д.), та оптимізують технологічні процеси його виготовлення. Тільки після отримання задовільних результатів приймається рішення про виготовлення фізичного об'єкта.

Комп'ютерне прототипування об'єкту та циклу його виробництва включає в себе:

1. концептуальне моделювання, тобто попередню розробку декількох варіантів виробу, внаслідок якої з'являються "трьохмірні ескізи";
2. створення комп'ютерних "рисуноків", що являють собою ортогональні проекції майбутнього виробу;
3. власне 3D моделювання: трасування рисунків, тобто створення трьохмірних сплайнів, а потім - побудова поверхонь по цим сплайнам;
4. задання матеріалів та оцінку властивостей об'єктів, що моделюються;
5. передачу 3-х мірної моделі об'єкта в САПР для подальшої розробки внутрішнього пристрою об'єкта, конструкторських розрахунків, оформлення креслярської документації і т.д.

Створений за допомогою систем моделювання об'єкт, можна вміщувати в різні середовища, імітувати і простежувати не тільки його переміщення в створеному для нього віртуальному просторі, але й демонструвати його функціонування.

Якщо віртуальну реальність використати просто як засіб комунікації між учасниками процесу проектування, вона дозволить проектувальникам, фахівцям з надійності систем, персоналу і іншим фахівцям обговорювати достоїнства і недоліки проекту. Використовуючи

Лекція 9. Технічне забезпечення процесу проектування
віртуальну модель як наочну, можна довільно переміщувати його в просторі, "переміщатись" по ньому і т.д. А це неминуче призводить до більш ясного розуміння суті проблем і більш швидкого вироблення рішень по усуненню потенційних ускладнень під час проектування і виробництва будь-якого виробу.

<навести приклади 3D моделювання деталей в сучасних САПР >

Визначення моделі

Модель (від лат. *modulus* - міра, зразок) - пристрій або образ (уявна або умовний: схема, креслення, система рівнянь і т.п.) якого-небудь об'єкта або явища (оригіналу даної моделі), що адекватно відбиває його досліджувані властивості й може використовуватись як заступник реального об'єкта в наукових або інших цілях (рис. 1).

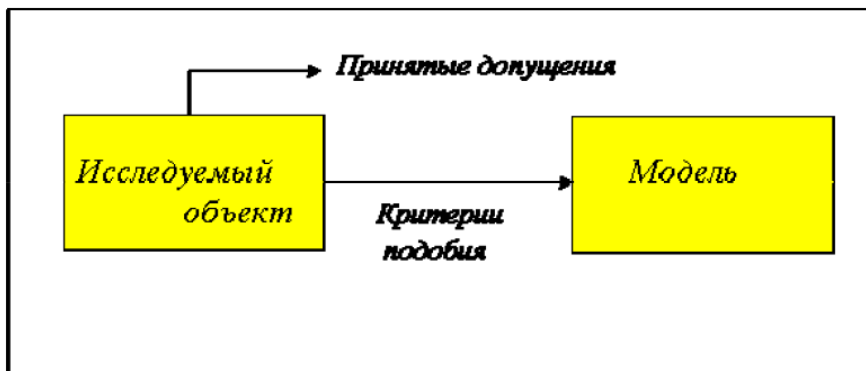


Рис. 1. Зв'язок між поняттями «Об'єкт» - «Модель»

Моделювання - це робота з прогнозування характеристик життєвого циклу продукту до його виробництва. Характеристики продукту включають витрати на його розробку, експлуатацію, видаткові матеріали (паливо), тривалість життя продукту, міцність, безпека, шум, надійність, комфортність, простота виготовлення й обслуговування, витрати на гарантійний ремонт, час виведення на ринок, прибутковість і багато чого іншого. Моделювання виконується за допомогою програмного забезпечення, здатного прогнозувати всі ці характеристики життєвого циклу продукту. Моделювання виконується для прогнозування майбутнього, а особливості продукту систематично варіюються з метою поліпшення характеристик продукту на ранньому етапі розробки, при цьому рішення про технологічне і матеріальне оснащення приймаються заздалегідь. Моделювання допомагає проектувальникам зрозуміти, які компроміси і рішення в області проекту вони повинні зробити для оптимізації своєї продукції. Моделювання скорочує і замінює дорогий тривалий процес створення фізичних прототипів і їхнього тестування. Моделювання може впливати на кожний з пунктів, позначених вище, приводячи Вас до збільшення продажів і скороченню витрат на розробку продукту.

Класифікація ММ СОС

Види математичних моделей:

1. За формою подання:
 - a. фізичні;
 - b. математичні;
 - c. аналогові;
 - d. цифрові.
2. По призначенню:

систем

- a. функціональні;
- b. структурні;
- c. геометричні;
- d. кінематичні;
- e. динамічні.

3. По методу дослідження:

- a. графічні;
- b. чисельні;
- c. графо-аналітичні;
- d. енергетичні;
- e. кінето-статичні;
- f. експериментальні.

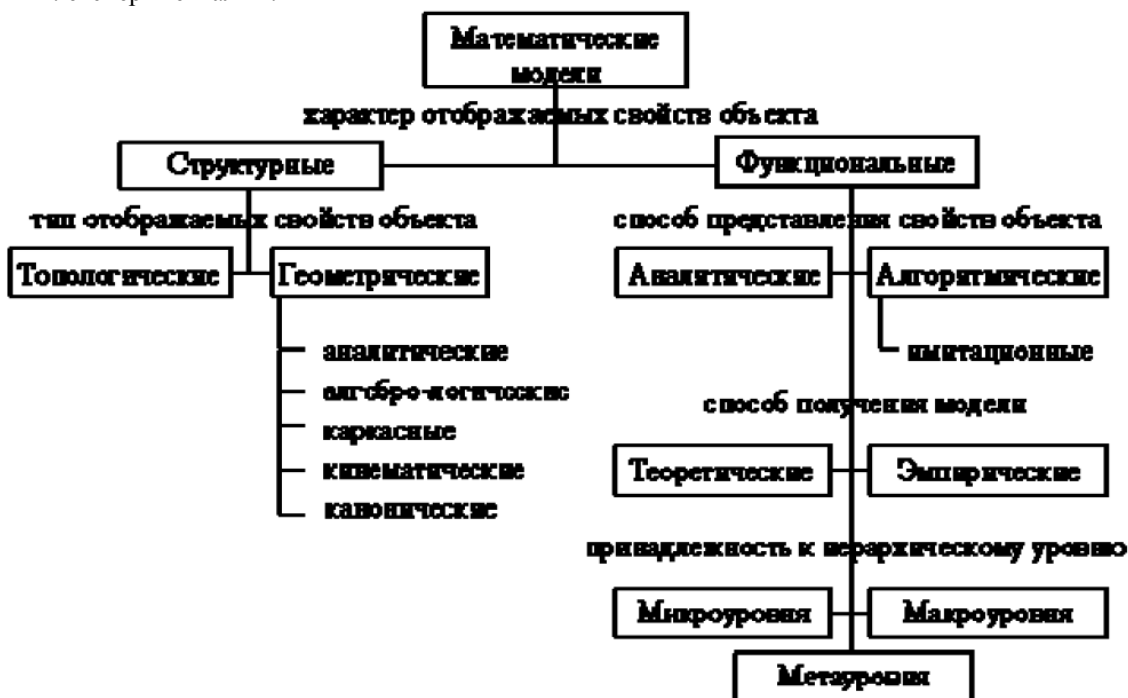


Рис.2. Типы математических моделей в АП СОС.

Структурні моделі призвані відобразити структурні властивості об'єкта.

Топологічні - для відображення складу і взаємозв'язку елементів. Мають форму графів, таблиць, матриць, списків. Їх використовують для рішення задач компонування устаткування, розміщення деталей, тощо, та при структурному синтезі.

Геометричні – забезпечують опис форми і взаємного розташування елементів у просторі. Застосовуються для рішення задач конструкторського аспекту. Використовують кілька типів геометричних моделей.

Для відображення геометричних властивостей порівняно простих поверхонь (до другого порядку) застосовують аналітичні й алгебро-логічні моделі.

Аналітичні моделі описують поверхні і лінії у вигляді рівнянь (аналітична геометрія), де окремим випадком є канонічні моделі. Канонічні моделі використовують у тих випадках, коли вдається виділити параметри, що цілком визначають об'єкт, і в той же час однозначно зв'язані з його формою. Наприклад для циліндра такими параметрами є направляючі косинуси осі, радіус і довжина циліндра, просторові координати точки осі, що належить одній з основ.

Алгебро-логічні застосовуються для опису тіл у виді систем аналітичних і логічних виразів, що відбивають умови приналежності точок внутрішнім областям тел.

Для відображення геометричних властивостей складних поверхонь застосовують каркасні і кінематичні моделі.

176Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Лекція 9. Технічне забезпечення процесу проектування

Каркасні моделі представляють поверхні тіл у виді каркасів - сукупності ліній, що утворюють сітку на поверхні тіла. Застосовують досить прості способи апроксимації сітки у виді кусочно-лінійної апроксимації, сплайн - апроксимації і т.п. за значеннями координат у вузлах сітки.

утворюючої), що рухається по направляючій лінії.

Функціональні моделі призначені для опису фізичних й інформаційних процесів, що відбуваються в об'єкті і являють собою взаємозв'язок між внутрішніми, зовнішніми, вихідними параметрами і фазовими змінними. По способі представлення властивостей об'єкта функціональні моделі поділяються на аналітичні й алгоритмічні.

Аналітичні моделі мають явну залежність вихідних параметрів від внутрішніх і зовнішніх, тобто мають вигляд (1). Аналітична модель може бути отримана з (2) у тому випадку, якщо така модель допускає аналітичне рішення. У протилежному випадку модель (2) повинна бути приведена до алгоритмічного виду.

Алгоритмічна модель – це вираження взаємозв'язку між вихідними, внутрішніми і зовнішніми параметрами у виді алгоритму - тобто послідовності обчислень, за допомогою яких вихідні дані перетворюються за допомогою ЕОМ у вихідні параметри. Для перетворення системи рівнянь (2) в алгоритмічну модель необхідно доповнити її алгоритмом згідно обраним методом чисельного рішення - тобто визначення вектора фазових змінних і алгоритма перетворення вектора фазових змінних у вектор вихідних параметрів.

Важливим варіантом алгоритмічної моделі є **імітаційна модель** - яка призначена для імітації (наслідування) фізичних та інформаційних процесів у досліджуваному об'єкті при заданих змінах у часі зовнішніх впливів на об'єкт. Прикладом імітаційних моделей можуть бути моделі поведінки динамічних об'єктів у виді систем диференціальних рівнянь, а також моделі систем масового обслуговування, призначені для імітації проходження заявок через систему.

За способом отримання моделей, вони розділяються на теоретичні й емпіричні (експериментальні).

Теоретичні моделі отримують в результаті вивчення фізичних закономірностей процесів, що протікають в об'єкті, обґрунтуванні і прийнятті припущень, що спростують, визначенні відповідного математичного опису, виконанні необхідних викладень і приведення моделі до прийнятої форми представлення.

Експериментальні моделі базуються на використанні зовнішніх проявів властивостей об'єкта, що фіксуються під час експлуатації або при проведенні цілеспрямованих експериментів. Відношення до об'єкта як до чорного ящика без з'ясування фізичних основ процесів, що відбуваються. Зазвичай такі моделі мають просту аналітичну форму.

Математичне моделювання на різних етапах проектування

Математичне моделювання СОС застосовується, як правило, для розрахунку і вибору оптимальних параметрів об'єкта або системи, за умови, що побудова математичної моделі можливо, і для умов статистики. Якщо побудова математичної моделі неможливо, то застосовують статистичні методи і методи нейро-мережевого підходу.

Моделювання також використовують для скорочення часу на кожному етапі життєвого циклу СОС.

Моделювання на етапі Концептуального Проектування.

На етапі розробки концепції, основне рішення на вищому рівні приймається виходячи з ринкових потреб у даному виробі, вимог функціональності продукту і нестатків бізнесу. Моделювання на цьому етапі дозволяє розроблювачам концепції перевірити неї, щоб переконатися, що продукт можна виготовити відповідно до заданих вимог і споживчих властивостей. Моделювання полегшує попередню оцінку різних концепцій проекту, надаючи можливість задоволення усіх вимог до функціональності у встановлених рамках часу і витрат засобів на виконання робіт з даного проекту.

Моделювання на етапі Детального Проектування.

систем

На цьому етапі проект у цілому уже визначений, включаючи його окремі вузли і компоненти з заданими споживчими властивостями. Визначено геометрію, матеріали і стадії виробничого процесу. Моделювання на цьому етапі дозволяє упевнитися, що даний проект реальний, його можна буде виконати і поставити на рейки серійного виробництва. Весь проект може бути змодельований, від системи в цілому до кожного окремого компонента. На цьому етапі моделювання виконується інженерами-розроблювачами й інженерами-технологами, що планують виробництво.

Моделювання на етапі Випробувань.

Це один з важливих етапів у житті проекту, тому що він визначає подальшу долю виробу. Етап випробувань починається тоді, коли доступний зразок. Більшість компаній створюють кілька натурних прототипів і піддають їх ретельному тестуванню. Якщо експериментальні зразки не проходять іспитів, проект змінюється і зразок проходить нові іспити, доти, поки не буде отриманий позитивний результат. Такий цикл створення, іспити й усунення дефектів вимагає значних тимчасових, виробничих і фінансових витрат. Моделювання може використовуватися на цій стадії для зменшення числа прототипів і фізичних циклів відновлення і повторного випробування, шляхом моделювання тестируемого зразка на комп'ютері. Якщо тестируемый зразок не проходить модельних іспитів, повторна зміна проекту дуже недорого. Використовуючи комп'ютерне моделювання на даному етапі вдається скоротити витрати на розробку прототипів більш ніж на 50%, а це значна економія часу і засобів. Крім цього, моделювання може бути використане для скорочення часу й обсягу іспитів натурального зразка. Наприклад, за допомогою моделювання можна «пробігтися» за зразком, повідомивши «дані», які б вимірялися при фізичному іспиті. Це дозволяє інженерам, що роблять іспит, заздалегідь і більш якісно визначати критерії вимірів і навантаження/динамічного навантаження, заощаджуючи час і крім «роботи наосліп» як часто буває при лабораторних дослідженнях.

Про значення моделювання на етапі Виробництва.

Усім зрозуміло, що на цьому етапі продукція виготовляється. У даному випадку моделювання використовується для оптимізації процесів виробництва з метою мінімізації відходів і етапів обробки. Моделювання поліпшує процедуру виготовлення продукту і знижує витрати на гарантійний ремонт, виявляючи і крім тих параметрів проекту, що можуть викликати дефекти під час процесу виробництва. Більш того, деякі з нових виробничих процесів, такі як гідро-формування і суперпластичне формування, вимагають моделювання для ідентифікації параметрів виробничого процесу, таких як температура, тиск і швидкість.

Про значення моделювання на етапі Маркетингу/Продажів.

Моделювання може використовуватися не голько для розробки і виробництва Вашої продукції. Воно також може застосовуватися для маркетингу і продажів Вашої продукції. Моделювання можна використовувати в комерційних пропозиціях, щоб показати, як Ваш продукт буде виглядати в конкретних умовах і яких цікавих властивостях він може бути цікавий покупцеві. Результати моделювання можуть використовуватися в рекламі для додання Вашому продуктові образа «высокотехнологичного». Ваші продажі збільшаться завдяки тому, що моделювання поліпшує дизайн, а також саму продукцію, тобто її споживчі властивості.

Про значення моделювання на етапі Підтримки і Супроводу.

Надалі, після реалізації, Ваша продукція буде потребувати підтримки, технічному обслуговуванні, капітальному і поточному ремонті. Моделювання може використовуватися на цьому етапі для проведення ремонту і для модифікацій виробу, забезпечуючи коректування проблеми, при цьому одночасно зберігаючи функціональність первісного дизайну і набір корисних властивостей. Задачі цього етапу дуже чітко проглядаються в аерокосмічній промисловості, де моделювання застосовується для продовження ресурсу життєвого циклу виробів.

Про значення моделювання на етапі Утилізації.

Це останній і теж досить важливий етап життєвого циклу виробів і технологій, особливо актуальний у даний час. Ніщо не вічно в цьому світі. Коли термін корисного використання виробу закінчений, він утилізується або переробляється. Моделювання тут застосовується для вибору таких виробничих процесів і пакувальних матеріалів, при яких можлива економічна переробка виробу, включаючи різні типи матеріалів, що були використані для його виготовлення. У багатьох галузях промисловості число «контактуючих частин» під час виробництва на порядок перевищує число фактичних частин у продукті. Застосування моделювання дозволяє спланувати їхнє повторне використання або ефективну утилізацію ще на етапі ескізного проектування самого виробу, значно задовго до початку його експлуатації.

А де ж повернення інвестицій, витрачених на моделювання? Коли Ви купуєте програмне забезпечення для

Лекція 9. Технічне забезпечення процесу проектування

моделювання, включаючи гарантійне обслуговування і комп'ютери, Ваша компанія робить інвестиції. Інвестиції - це витрати, що, по ідеї, повинні повертатися. Таким чином, інвестиція дає дохід (фінансовий прибуток), що по розмірі повинний звичайно перевершувати саму інвестицію. Тому в першу чергу Вас буде цікавити «прибутковість інвестицій», існує і такий термін. Давайте тепер звернемося до розгляду й аналізу прибутковості інвестицій у моделюючі комп'ютерні технології. Існує кілька аспектів «прибутковості інвестицій» при вкладенні засобів у моделюючі технології:

- Вид інвестиції (разова або постійна);
- Валовий прибуток;
- Доход від інвестиції (фінансовий прибуток);
- Внутрішня норма прибутку.

Термін окупності інвестицій.

Тепер визначивши з теорією по питанню про користь комп'ютерного моделювання, ми готові перейти до сугубо практичних питань і порахувати на конкретних прикладах, реальні гроші в карбованцях або в умовних одиницях, що у виді прибутку вже одержали деякі російські підприємства, що впровадили й активно використовують моделюючі комп'ютерні технології.

14. Компоненти математичного забезпечення (МЗ) АПСОС

Поняття «математичного забезпечення» в АП СОС

Математичним забезпеченням САПР згідно ГОСТ 27487-97 (СРСР) являється сукупність математичних методів, моделей і алгоритмів, необхідних для автоматизованого проектування, які представлені в заданій формі.

Процес автоматизованого проектування можна представити як процес перетворення математичних моделей. Тому характеристики САПР в основному залежать від властивостей реалізованого в них математичного забезпечення. А самі математичні моделі служать для опису властивостей об'єктів в процедурах автоматизованого проектування.

Під математичною моделлю розуміють систему математичних об'єктів (чисел, змінних, матриць, множин і т.д.) та відношень між ними, що відбивають, важливі для проектування, властивості технічного об'єкта.

Прикладом узагальненого запису математичної моделі СОС може служити:

$$Y = F(X, Q) \quad (1)$$

де Y, X, Q - відповідно вектори вихідних, внутрішніх і зовнішніх параметрів; F - деяка відома функція, розмірність якої збігається з розмірністю вектора вихідних параметрів $F = \{f_1, \dots, f_n\}$. У даному випадку існує явний функціональний зв'язок між вихідними параметрами системи (тобто властивостями системи) і її внутрішніми і зовнішніми параметрами. По такій явній залежності легко оцінювати вплив внутрішніх і зовнішніх параметрів на вихідні параметри системи. Однак у більшості випадків у явному виді математичну модель створити не вдається.

Крім внутрішніх, зовнішніх і вихідних параметрів виділяють т.зв. фазові змінні $V = v(W, t)$. Фазові змінні характеризують фізичний або інформаційний стан об'єкта, а їх зміну в часі виражають перехідні процеси в об'єкті. Фазові змінні в загальному випадку є функціями незалежних координат $W = \{w_1, \dots, w_n\}$ і часу t . Прикладом фазових змінних у моделі гідроциліндра можуть служити тиск і витрата рідини для гідравлічної, та швидкість і зусилля на поршні для механічної підсистем.

У загальному випадку математична модель об'єкта може бути представлена в наступному виді:

$$\Psi(w_i, t, v, \partial v / \partial t, \partial^2 v / \partial t^2, \partial v / \partial w_i, \partial^2 v / \partial w_i^2) = 0, \quad (2)$$

або в скороченій формі

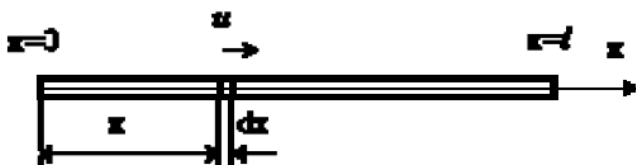
$$Lv(W, t) = \phi(W, t) \quad (2')$$

Тут L - вектор операторів над фазовими змінними ∂, \int - диференціальні, інтегральні оператори, алгебраїчні функції; $v(W, t)$ - шукана функція (фазова змінна); ϕ - задана вектор - функція.

Прикладом запису математичної моделі в загальному вигляді може служити одномірне хвильове рівняння, що описує подовжні коливання стрижня з розподіленою масою (рис.3):

$$\frac{\partial^2 u}{\partial x^2} - \frac{1}{a^2} \times \frac{\partial^2 u}{\partial t^2} = 0$$

$L = \frac{\partial^2}{\partial x^2} - \frac{1}{a^2} \times \frac{\partial^2}{\partial t^2}$; $a = \sqrt{E/\rho}$ - швидкість поширення звуку в матеріалі стрижня; $W = x$ - незалежна змінна - координата точки уздовж осі стрижня; $V = u(x)$ - фазова (залежна) змінна - переміщення точок стрижня уздовж осі, $\phi = 0$.



Лекція 9. Технічне забезпечення процесу проектування

Рис.3. Подовжні коливання стрижня з розподіленою масою.

Більшість вихідних параметрів Y є функціоналами фазових змінних. Для їх визначення спочатку необхідно при заданих зовнішніх і внутрішніх параметрах розв'язати систему рівнянь - тобто знайти фазові змінні, а вже потім визначити вихідні параметри системи. Для нашого випадку вихідними змінними будуть напруги в стрижні, що залежать від величини деформацій, а ті, у свою чергу, від величини переміщень.

Під функціоналом розуміють відображення безлічі функцій на безліч чисел, коли кожній функції (з деякої безлічі) ставиться у відповідність яке-небудь число. У тому випадку, якщо безліч функцій складається з констант, то поняття функціонала збігається з поняттям функції. Іншим прикладом функціонала є інтеграл

$$J(x) = \int_a^b x(t) dt$$

, де $x(t)$ - неперервна функція, визначена на відрізку $[a, b]$.

Прикладами таких вихідних параметрів - функціоналів є к.к.д, потужність, довговічність, амплітуда коливань і т.п.

Ієрархія математичних моделей СОС

Використання принципів блочно-ієрархічного підходу до проектування приводить до ієрархії математичних моделей проєктованих об'єктів по рівнях абстракції. Об'єднання рівнів, споріднених по характеру використовуваного математичного апарата, приводить до утворення трьох укрупнених рівнів:

- мікрорівень (моделі з розподіленими параметрами)
- макрорівень (моделі з зосередженими параметрами)
- метарівень (системний рівень)

Різні рівні абстракції відрізняються один від одного ступенем деталізації опису властивостей елементів об'єкта, формою представлення моделі фазовими і незалежними змінними, методами рішення. Як правило, від мікро- до мета- рівня збільшується ступінь деталізації опису і розмірність математичної моделі. У загальному випадку, на мікрорівні маємо - нескінченне число ступенів свободи.

Мікрорівень являє собою найбільш детальний розгляд властивостей об'єкта, що представляється у виді неперервного суцільного середовища. Точне рішення математичної моделі в більшості випадків є неможливим, тому будують наближену дискретну модель. СОС, отримані при дискретизації, є системами алгебраїчних рівнянь великих порядків. З ростом складності задачі приходять до

необхідності введення допущень, що спрощують математичну модель.

На **мікрорівні** типові математичні моделі (ММ) представлені диференціальними рівняннями в часткових похідних (ДРЧП) разом із крайовими умовами. До цих моделей, які називають розподіленими, відносять рівняння математичної фізики. Об'єктами дослідження тут являються поля фізичних величин, що є необхідним при аналізі міцності будівельних конструкцій або деталей машин, при дослідженні процесів в рідких середовищах, моделюванні концентрацій і потоків частинок, тощо.

Число спільно досліджуваних різних середовищ (число деталей, шарів матеріалу, фаз агрегатного стану) у практично використовуваних моделях мікрорівня не може бути великим через складності обчислювального характеру. Різко знизити обчислювальні витрати в багатокomпонентних середовищах можна, тільки застосувавши інший підхід до моделювання, заснований на прийнятті визначених допущень.

Макрорівень використовує представлення про об'єкт, як про дискретний простір. Зокрема, для механічних систем інерційні параметри вважаються зосередженими у визначених точках або перетинах системи, що зв'язані між собою пружно-диссипативними або геометричними зв'язками. Елементами цього рівня стають об'єкти, що на мікрорівні розглядалися як системи. З ростом числа елементів розмірність задачі зростає і стає необхідним перехід до наступного ієрархічного рівня.

систем

Моделями макрорівня, названими також *зосередженими*, є системи алгебраїчних і звичайних диференціальних рівнянь, оскільки незалежної змінної тут залишається тільки час t . Спрощення опису окремих компонентів (деталей) дозволяє досліджувати моделі процесів у пристроях, приладах, механічних вузлах, число компонентів у які може доходити до декількох тисяч.

У тих випадках, коли число компонентів у досліджуваній системі перевищує деякий поріг, складність моделі системи на макрорівні знову стає надмірною. Тому, приймаючи відповідні допущення, переходять на *функціонально-логічний рівень*. На цьому рівні використовують апарат передатних функцій для дослідження аналогових (безперервних) процесів або апарат математичної логіки і кінцевих автоматів, якщо об'єктом дослідження є дискретний процес, тобто процес з дискретною безліччю станів.

Метарівень використовує дискретне представлення про простір і про час. Об'єктами дослідження є складні пристрої і комплекси, функціонування яких розглядається як ланцюг подій, що відбуваються в дискретні моменти часу i , що полягають у зміні стану елементів. Роль елементів виконують системи макрорівня.

Для дослідження ще більш складних об'єктів, ніж на мікрорівні, прикладами яких можуть служити виробничі підприємства і їхні об'єднання, обчислювальні системи і мережі, соціальні системи й інші подібні об'єкти, застосовують апарат теорії масового обслуговування, можливі використання і деякі інші підходи, наприклад, мереж Петри. Ці моделі відносяться до *системного* рівня моделювання.

Вимоги до ММ СОС

До математичних моделей, які використовують у САПР, висувають наступні основні вимоги:

- *високий ступінь універсальності* (тобто розширення моделі на більш широкий клас об'єктів);
- *достатня точність одержуваних результатів*;

(Модель завжди лише приблизно відбиває деякі властивості об'єкта. *Адекватність* має місце, якщо модель відбиває задані властивості об'єкта з прийнятною точністю. Під *точністю* розуміють ступінь відповідності оцінок однойменних властивостей об'єкта і моделі.)

- *максимальна економічність моделей у витраті обчислювальних ресурсів при їхній реалізації*;

(Економічність (обчислювальна ефективність) визначається витратами ресурсів, необхідних для реалізації моделі. Оскільки в САПР використовуються математичні моделі, далі мова йтиме про характеристики саме математичних моделей, і економічність буде характеризуватися витратами машинних часу і пам'яті.)

- *адекватність*;

Адекватність оцінюється переліком відбиваних властивостей і областями адекватності. *Область адекватності* — область у просторі параметрів, у межах якої погрішності моделі залишаються в припустимих межах. Наприклад, область адекватності лінеаризованої моделі поверхні деталі визначається системою нерівностей

де

— i -та координата j -ї точки поверхні в об'єкті і моделі відповідно, ε_{ij} і $\varepsilon_{дон}$ — допущена і гранично припустима відносні похибки моделювання поверхні, максимум береться по всіх координатах і контрольованих точках. В більшості випадків області адекватності будуються в просторі зовнішніх змінних. Так, область адекватності моделі електронного радіоелемента звичайно виражає припустимі для застосування моделі діапазони зміни температур, зовнішніх напруг, частот, тощо.

- *надійність*.

Ці вимоги суперечливі. Високий ступінь універсальності і достатня точність досягаються за рахунок ускладнення моделі, що погіршує економічність. Збільшення універсальності також приводить до можливості появи відмовлень через можливе неврахування якихось специфічних факторів - тобто знижується надійність. Найкраще компромісне задоволення цих вимог досягається за рахунок

Лекція 9. Технічне забезпечення процесу проектування

застосування широкого спектра математичних моделей, вибір яких залежить від особливостей розв'язуваних задач.

Місце процедур формування ММ у маршрутах проектування.

Процес проектування СОС на етапах аналізу складається включає в себе процедур формування моделі та її дослідження (рішення). У свою чергу, формування моделі включає дві процедури:

по-перше, розробку моделей окремих компонентів,
по-друге, формування моделі системи з моделей компонентів.

Перша з цих процедур виконується попередньо для типових компонентів СОС поза маршрутом проектування конкретних об'єктів. Як правило, моделі компонентів розробляються фахівцями в прикладних областях, причому з врахуванням вимог до моделей і форм їхнього представлення в САПР. Звичайно, на допомогу розробникам САПРівських моделей пропонуються методики і допоміжні засоби, наприклад, у виді програм аналізу для експериментального відпрацювання моделей. Створені моделі включаються в бібліотеки моделей прикладних програм аналізу.

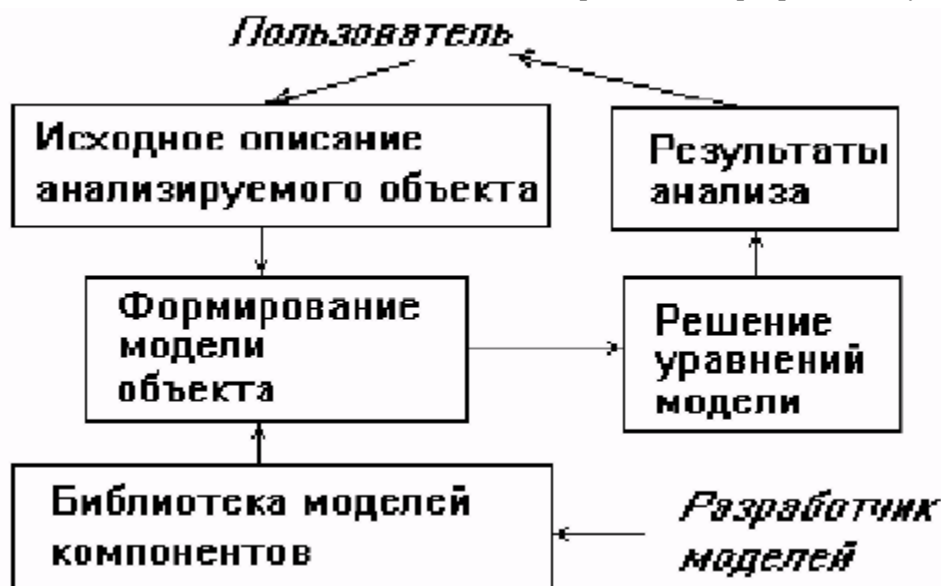


Рис.4. Місце процедур формування моделей на маршрутах проектування.

На маршруті проектування кожного нового об'єкта виконується нова процедура (рис. 4) формування моделі системи з використанням бібліотечних моделей компонентів. Як правило, ця процедура виконується автоматично по алгоритмах, включених у заздалегідь розроблені програми аналізу. Приклади таких програм існують в різних додатках і насамперед у галузях загального машинобудування, приладобудування та радіоелектроніки.

При застосуванні цих програм користувач описує досліджуваний об'єкт вхідною мовою програми аналізу не у вигляді системи рівнянь, а у вигляді списку елементів структури, еквівалентної схеми, ескізу або креслення конструкції. А система рівнянь буде сформована та отримана програмою автоматично.

15. ММ в процедурах аналізу на макрорівні

Загальна характеристика.

Математичний опис процесів в об'єктах на макрорівні представляють системами звичайних диференціальних і алгебраїчних рівнянь. Аналітичні рішення таких систем при типових значеннях їхніх порядків у практичних задачах отримати не вдається, тому в САПР переважно використовуються алгоритмічні моделі. У цьому параграфі викладений узагальнений підхід до формування алгоритмічних моделей на макрорівні, справедливий для більшості додатків.

Вихідними для формування математичних моделей об'єктів на макрорівні є компонентні і топологічні рівняння.

Компонентними рівняннями називають рівняння, що описують властивості елементів (компонентів), іншими словами, це рівняння *математичних моделей елементів* (ММЕ).

Топологічні рівняння описують взаємозв'язки в системі, що моделюється.

У сукупності, компонентні і топологічні рівняння конкретно СОС являють собою вихідну *математичну модель*.

Очевидно, що компонентні і топологічні рівняння в системах різної фізичної природи відбивають різні фізичні властивості, але можуть мати однаковий формальний вид. Однакова форма запису математичних співвідношень дозволяє говорити про *формальні аналогії* компонентних і топологічних рівнянь. Такі аналогії існують для механічних поступальних, механічних обертальних, електричних, гідравлічних (пневматичних), теплових об'єктів. Наявність аналогій приводить до практично важливого висновку: *значна частина алгоритмів формування і дослідження моделей у САПР виявляється інваріантною і може бути застосована до аналізу проєктованих об'єктів у різних предметних областях*. Єдність математичного апарата формування ММ СОС є особливо зручним при аналізі систем, що складаються з фізично різнорідних підсистем.

У перерахованих вище прикладах компонентні рівняння мають вигляд

(3)

і топологічні рівняння

(4)

де - вектор фазових змінних, t - час.

Розрізняють фазові змінні двох типів: 1)- змінні типу потенціалу (наприклад, електрична напруга) і 2)- змінні типу потоку (наприклад, електричний струм). Кожне компонентне рівняння характеризує зв'язок між різнотипними фазовими змінними, стосовно одного компонента (наприклад, закон Ома описує зв'язок між напругою і струмом у резисторі, а топологічне рівняння зв'язку між однотипними фазовими змінними в різних компонентах.)

Моделі можна представляти у виді систем рівнянь або в графічній формі, якщо між цими формами установлена взаємно однозначна відповідність. Як графічну форму часто використовують еквівалентні схеми.

Приклади компонентних і топологічних рівнянь.

Електричні системи.

В електричних системах фазовими змінними являються електричні напруження і струми. Компонентами систем можуть бути прості двохполюсні елементи і більш складні двох- і багатопольюсні компоненти. До простих 2-полюсників відносяться наступні елементи: резистор, ємність, індуктивність, що характеризуються одноіменними параметрами R , C , L . В еквівалентних схемах ці елементи позначають відповідно до рис. 5,а.

Компонентні рівняння простих двохполюсників:

для R : $i = i R$ (закон Ома), (5)

для C : $i = C du/dt$, (6)

для L : $u = L di/dt$, (7)

де u - напруга (точніше, спадання напруги на двохполюснику); i - струм.

Лекція 9. Технічне забезпечення процесу проектування

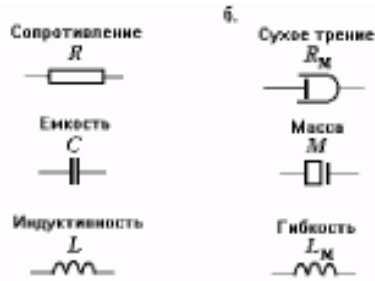


Рис. 5. Умовні позначення простих елементів в еквівалентних схемах: а) електричних, гідравлічних, теплових; б) механічних.

Ці моделі лежать в основі моделей інших можливо більш складних компонентів. Велика складність може визначатися нелінійністю рівнянь (5) - (7) (тобто залежністю R , C , L від фазових змінних), або обліком залежностей параметрів R , C , L від температури, або наявністю більш двох полюсів. Однак багатополісні компоненти можуть бути зведені до сукупності взаємозалежних простих елементів.

Топологічні рівняння виражають закони Кірхгофа для напруг (ЗКН) і струмів (ЗКС). Згідно ЗКН, сума напруг на компонентах уздовж будь-якого замкнутого контуру в еквівалентній схемі дорівнює нулеві, а відповідно до ЗКС сума струмів у будь-якому замкнутому перетині еквівалентної схеми дорівнює нулеві:

$$\sum_{k \in K_p} u_k = 0,$$

$$\sum_{j \in J_q} i_j = 0,$$

(8) (9)

де K_p - множина номерів елементів p -го контуру, J_q - безліч номерів елементів, що входять у q -й перетин.

Прикладом ММ складного компонента може служити модель транзистора. На рис. 5 представлена еквівалентна схема біполярного транзистора, на якій залежні від напруг джерела струму $i_{эд} = i_{тэ} \exp(u_{э}/(m\phi T))$ і $i_{кд} = i_{тк} \exp(u_{к}/(m\phi T))$ відображають статичні вольтамперні характеристики р-п переходів, $i_{тэ}$ й $i_{тк}$ — теплові струми переходів, $m\phi T$ — температурний потенціал, $u_{э}$ і $u_{к}$ — напруги на еміттерному і колекторному переходах, $C_{э}$ і $C_{к}$ — ємності переходів, $R_{уэ}$ і $R_{ук}$ — опору витоку переходів, $R_{б}$ і $R_{к}$ — об'ємні опори тіл бази і колектора, $i_{г} = V_{иэд} - V_{иікд}$ — джерело струму, що моделює підсилювальні властивості транзистора, I і $I_{и}$ — прямий і інверсний коефіцієнти підсилення струму бази. Тут $u_{э}$, $u_{к}$, $i_{эд}$, $i_{кд}$, $i_{г}$ — фазові змінні, а інші величини — параметри моделі транзистора.

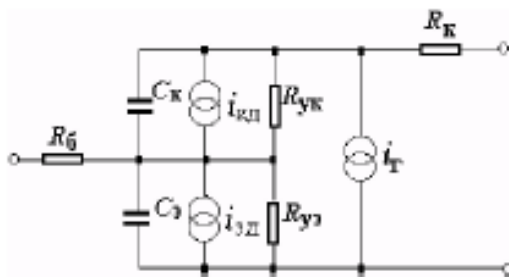


Рис. 6. еквівалентна схема біполярного транзистора.

Механічні системи.

Фазовими змінними в механічних поступальних системах є сили і швидкості. Використовують одну з двох можливих електромеханічних аналогій. Надалі будемо використовувати ту з них у якій швидкість відносять до фазових змінних типу потенціалу, а силу вважають фазовою змінною типу

систем

потіку. З огляду на формальний характер подібних аналогій, аналогічно можна застосовувати і протилежну термінологію.

Компонентне рівняння, що характеризує інерційні властивості тіл, у силу другого закону Ньютона має вигляд

$$(10) \quad \text{де } F - \text{ сила; } M - \text{ маса; } u - \text{ поступальна швидкість.}$$

Пружні властивості тіл описуються компонентним рівнянням, яку можна одержати з рівняння закону Гука. В одномірному випадку (якщо розглядаються подовжні деформації пружного стержня)

$$(11) \quad \text{де } G - \text{ механічна напруга; } E - \text{ модуль пружності; } \Delta l - \text{ відносна деформація, } \Delta l - \text{ зміна довжини } l \text{ пружного тіла під впливом } G. \text{ З огляду на те, що } G = F/S, \text{ де } F - \text{ сила, } S - \text{ площа поперечного переріза тіла, і диференціюючи (11), маємо}$$

$$(12) \quad \text{де } g = (SE/l) - \text{ твердість (величину, зворотну твердості, називають гнучкістю LM); } i = d(\Delta l)/dt - \text{ швидкість.}$$

Дисипативні властивості в механічних системах твердих тіл виражаються співвідношеннями, що характеризують зв'язок між силою тертя і швидкістю взаємного переміщення тертьових тіл, причому в цих співвідношеннях похідні сил або швидкостей не фігурують, як і у випадку опису за допомогою закону Ома дисипативних властивостей в електричних системах.

Топологічні рівняння характеризують, по-перше, закон рівноваги сил: сума сил, прикладених до тіла, включаючи силу інерції, дорівнює нулеві (принцип Даламбера), по-друге, закон швидкостей, відповідно до якого сума відносної, переносної й абсолютної швидкостей дорівнює нулеві.

У механічних обертальних системах справедливі компонентні і топологічні рівняння поступальних систем із заміною поступальних швидкостей на кутові, сил на обертальні моменти, мас – на моменти інерції, твердостей – на обертальні твердості.

Умовні позначки простих елементів механічної системи показані на рис. 4(б).

Неважко помітити наявність аналогій між електричною і механічною системами. Так, струмам і напругам у першій з них відповідають сили (або моменти) і швидкості механічної системи, компонентним рівнянням (6) і (7) і параметрам, що фігурує в них, C і L — рівняння (10) і (12) і параметри M і L_m , очевидні аналогія і між топологічними рівняннями. Далі параметри C і L будемо називати ємнісними (ємнісного типу), параметри L і L_m — індуктивними (індуктивного типу), а параметри R і $R_{tr} = \partial u / \partial F$ — резистивними (резистивного типу).

Мається й істотна відмінність у моделюванні електричних і механічних систем: перші з них одномірні, а процеси в других часто приходиться розглядати в двох- (2D) або тривимірному (3D)

просторі. Отже, при моделюванні механічних систем у загальному випадку в просторі 3D потрібно використовувати векторне представлення фазових змінних, кожна з яких має шість складових, відповідних шести ступенів волі.

Однак відзначені вище аналогії залишаються справедливими, якщо них відносити до проєкцій сил і швидкостей на кожен просторову вісь, а при графічному представленні моделей використовувати шість еквівалентних схем — три для поступальних складових і три для обертальних.

Гідравлічні системи.

Фазовими змінними в гідравлічних системах є витрати і тиски. Як і в попередньому випадку, компонентні рівняння описують властивості рідини розсіювати або накопичувати енергію.

Розглянемо компонентні рівняння для рідини на лінійній ділянці трубопроводу довжиною l і скористаємося рівнянням Нав'є-Стокса в наступній його формі (для ламінарного плину рідини)

$$(3.11) \quad \text{де } \eta - \text{ щільність рідини; } U - \text{ швидкість; } P - \text{ тиск; } a - \text{ коефіцієнт лінеаризованного грузлого тертя. Тому що } U = Q/S, \text{ де } Q - \text{ об'ємна витрата; } S - \text{ площа поперечного переріза трубопроводу, то, заміняючи просторову похідну відношенням скінчених різниць, маємо}$$

$$(3.11)$$

Лекція 9. Технічне забезпечення процесу проектування

де ΔP — падіння тиску на розглянутій ділянці трубопроводу. $L_r = \Delta l/S$ — гідравлічна індуктивність, що відбиває інерційні властивості рідини, $R_r = 2a/\rho$ — гідравлічний опір, що відбиває грузле тертя.

Примітка. У трубопроводі круглого перетину радіусом r зручно використовувати вираз для гідравлічного опору при ламінарному плинні: $R_r = 8\nu\Delta l/(\pi r^4)$, де ν — кінематична в'язкість; у випадку турбулентного характеру плинну рідини компонентне рівняння для грузлого тертя має вигляд $\Delta P = R_r Q/|Q|$ при $R_r = 0,37(\pi\nu^2/|Q|)^{1/4}$.

Інтерпретація рівняння (13) приводить до еквівалентної схеми рис.7.



Рис. 7. Еквівалентна схема ділянки трубопроводу

Явище стискальності рідини описується компонентним рівнянням, що випливає з закону Гука:

Диференціюючи (14) і з огляду на те, що об'ємна витрата Q зв'язана зі швидкістю $U = d(l)/dt$ співвідношенням $Q = U S$, одержуємо

де $C_2 = E/(\Delta S l)$ — гідравлічна ємність.

Зв'язок підсистем різної фізичної природи. Використовують наступні способи моделювання взаємозв'язків підсистем: за допомогою трансформаторно, гіраторних зв'язків і за допомогою залежності параметрів компонентів однієї підсистеми від фазових змінних іншої. В еквівалентних схемах трансформаторні і гіраторні зв'язки представлені залежними джерелами фазових змінних, показаними на рис. __. На цьому малюнку k і n — коефіцієнти трансформації; g — передатна провідність; U_j і I_j — фазові змінні в j -й ланцюга; $j=1$ відповідає первинної, а $j=2$ — вторинного ланцюга.

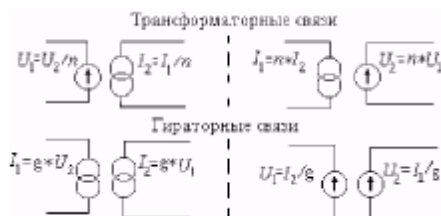


Рис. 8. Елементи взаємозв'язку підсистем різної фізичної природи

Представлення топологічних рівнянь.

Відомий ряд методів формування ММС на макрорівні. Одержувані з їхньою допомогою моделі розрізняються орієнтацією на ті або інші чисельні методи рішення і набором **базисних змінних**, тобто фазових змінних, що залишаються в рівняннях підсумкової ММС. Загальним для всіх методів є вихідна сукупність топологічних і компонентних рівнянь (3)-(4).

При записі топологічних рівнянь зручно використовувати проміжну графічну форму представлення моделі у виді еквівалентної схеми, що складає з двухполюсних елементів. Спільність підходу при цьому зберігається, тому що будь-який багатопольсний компонент можна замінити підсхемою із двухполюсників. У свою чергу еквівалентну схему можна розглядати як спрямований граф, дуги якого відповідають галузям схеми. Напрямку потоків у галузях вибираються довільно (якщо реальний напрямок при моделюванні виявиться протилежним, то це приведе лише до негативних чисельних значень потоку).

Приклад деякої простої еквівалентної схеми і відповідного їй графа приведена на рис. 9. Для конкретності і простоти викладу на рис. 3.6 використані умовні позначки, характерні для електричних еквівалентних схем, по тій же причині далі в цьому параграфі часто застосовується електрична термінологія. Очевидно, що пояснені вище аналогії дозволяють при необхідності легко перейти до позначень і термінів, звичним для механіків.

Для одержання топологічних рівнянь усі гілки еквівалентної схеми розділяють на підмножини хорд і гілок дерева. Мається на увазі **покриваюче** (фундаментальне) дерево, тобто підмножина з $\beta-1$

систем

дуг, не утворююче жодного замкнутого контуру, де β число вершин графа (вузлів еквівалентної схеми). На рис. 8(б) показаний граф еквівалентної схеми рис. 8(а), товстими лініями виділено одне з можливих покриваючих дерев.

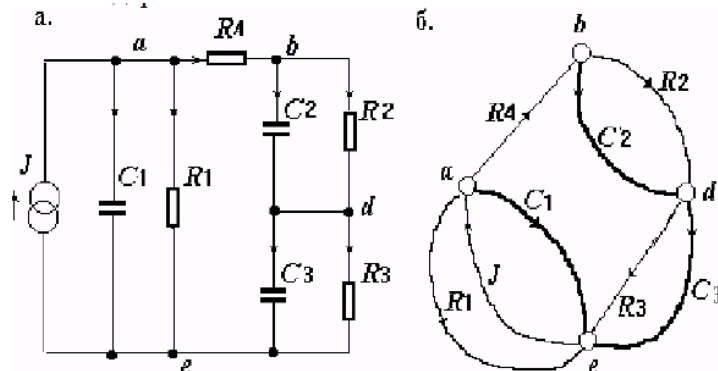


Рис. 9. еквівалентна схема (а) і її граф (б)

Вибір дерева однозначно визначає вектора напруг U_X і струмів I_X хорд, напруг $U_{ВД}$ і струмів $I_{ВД}$ гілок дерева і приводить до запису топологічних рівнянь у виді

$$U_X + M U_{ВД} = 0, \quad (13)$$

$$I_{ВД} - M^T I_X = 0, \quad (14)$$

де M — матриця контурів і перетинів. M^T - транспонована M -матриця.

Хорды	Ветви дерева		
	C1	C2	C3
R1	-1	0	0
R2	0	-1	0
R3	0	0	-1
R4	-1	+1	+1
J	+1	0	0

У M -матриці число рядків відповідає числу хорд, число стовпців дорівнює числу гілок дерева. M -матриця формується в такий спосіб. По черзі до дерева підключаються хорди. Якщо при підключенні до дерева p -ї хорди q -я гілка входить у контур, що утворився, то **елемент** M_{pq} матриці дорівнює +1 при збігу напрямків гілки і підключеної хорди; $M_{pq} = -1$ при розбіжності напрямків. У протилежному випадку $M_{pq} = 0$. Для схеми на рис. 8 M -матриця представлений у виді табл. 1.

Особливості еквівалентних схем механічних об'єктів.

Для кожного ступеня свободи будують свою еквівалентну схему. Кожному тілу з масою, що враховується, відповідає вузол схеми (вершина графа). Один вузол, називаний базовим, приділяється тілу, що ототожнюється з інерціальною системою відліку.

Кожен елемент маси зображують гілку, що з'єднує вузол відповідній масі тіла з базовим вузлом; кожен елемент пружності гілку, що з'єднує вузли тіл, зв'язаних пружним зв'язком; кожен елемент тертя — гілку, що з'єднує вузли тертьових тіл. Зовнішні впливи моделюються джерелами сил і швидкостей.

Як приклад на рис. 10(а) зображена деяка механічна система — візок, що рухається по дорозі і

Лекція 9. Технічне забезпечення процесу проектування складається з платформи A , коліс Y_1, Y_2 і ресор Z_1 і Z_2 . На рис. 10(б) приведена еквівалентна схема для вертикальних складових сил і швидкостей, на якій тілам системи відповідають однойменні вузли, враховуються маси платформи і коліс, пружність ресор, тертя між колісьми і дорогою; нерівності дороги викликають вплив на систему, зображене на рис. 10(б) джерелами сили.

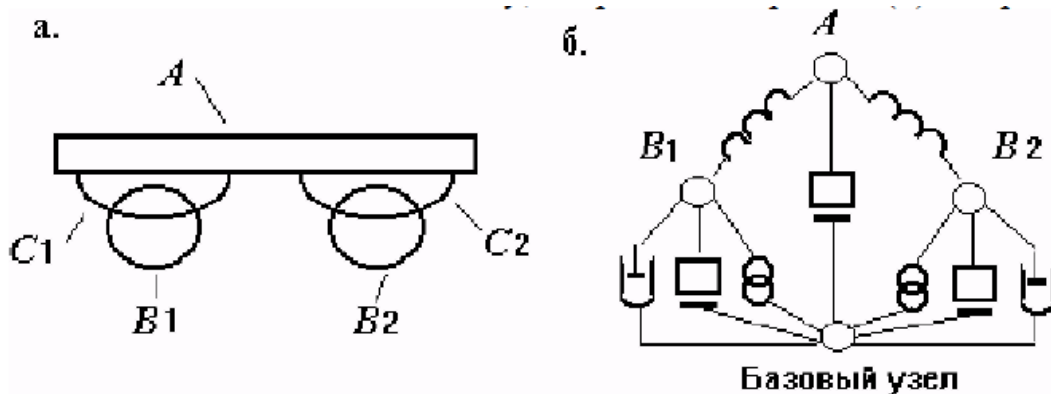


Рис. 10. Проста механічна система:
а) ескізне зображення;
б) еквівалентна схема.

Характеристика методів формування ММС.

Вихідну систему компонентних і топологічних рівнянь (3) і (4) можна розглядати як остаточну ММС, що і підлягає чисельному рішення. Чисельне рішення цієї системи рівнянь припускає *алгебралізацію* диференціальних рівнянь, наприклад, за допомогою перетворення Лапласа або формул чисельного інтегрування. У програмах аналізу нелінійних об'єктів на макрорівні, як правило, застосовуються формули чисельного інтегрування, прикладом яких може служити неявна формула Ейлера

де V_i значення змінної V на i -му кроці інтегрування; Δt - крок інтегрування. Алгебралізація має на увазі попередню дискретизацію незалежної змінної t (замість безперервної

змінної t одержуємо кінцеву безліч значень t_n). вона полягає в представленні ММС у виді системи рівнянь

(17)

з невідомими V_n і Z_n , де використане позначення $Z = dV/dt$. Цю систему алгебраїчних рівнянь, у загальному випадку нелінійних, необхідно вирішувати на кожному кроці чисельного інтегрування вихідних диференціальних рівнянь.

Однак порядок цієї системи досить високий і приблизно дорівнює $2\alpha + \gamma$, де α — число гілок еквівалентної схеми (кожна галузь дає дві невідомі величини фазові змінні типу потоку і типу потенціалу, за винятком гілок зовнішніх джерел, у кожній з яких невідома лише одна фазова змінна), γ — число елементів у векторі похідних. Щоб знизити порядок системи рівнянь і тим самим підвищити обчислювальну ефективність ММС, бажано виконати попереднє перетворення моделі (у символічному виді) перед її багатокроковим чисельним рішенням. Попереднє перетворення зводиться до виключення із системи частини невідомі і відповідні числа рівнянь. Залишкові невідомі називають *базисними*. У залежності від набору базисних невідомих розрізняють кілька методів формування ММС.

Відповідно до *методу змінних станів* (більш повна назва методу метод змінних, що характеризують стан), вектор базисних змінних W складається в *змінні стани*. Цей вектор включає ненадлишкову безліч змінних, що характеризують накопичену в системі енергію. Наприклад, такими змінними можуть бути швидкості тіл (кінетична енергія визначається швидкістю, тому що дорівнює $M\dot{l}^2$) ємнісні напруги, індуктивні струми і т.п. Очевидно, що число рівнянь не перевищує γ . Крім того, підсумкова форма ММС виявляється наближеної до явної форми представлення системи диференціальних рівнянь, тобто до форми, у якій вектор d/dt явно виражений через вектор W , що спрощує подальше застосування явних методів чисельного інтегрування. Метод реалізується шляхом особливого вибору системи хорд і гілок дерева при формуванні топологічних рівнянь. Оскільки явні

систем

методи чисельного інтегрування диференціальних рівнянь не знайшли широкого застосування в програмах аналізу, то метод змінні стани також втрачає актуальність і його застосування виявляється досить рідким.

У класичному варіанті *вузлового методу* в якості базисних змінних використовуються *вузлові потенціали* (тобто швидкості тіл відносно інерціальної системи відліку, абсолютні температури, перепади тиску між модельованою і зовнішнім середовищем, електричні потенціали щодо базового вузла). Число вузлових потенціалів і відповідно рівнянь у ММС виявляється рівним $\beta - 1$, де β - число вузлів в еквівалентній схемі. Звичайно β помітно менше α , отже, порядок системи рівнянь у ММС знижений більш ніж у два рази в порівнянні з порядком вихідної системи.

Однак класичний варіант вузлового методу має обмеження на застосування і тому в сучасних програмах аналізу найбільше поширення одержав *модифікований вузловий метод*.

16. Математичне забезпечення аналізу на мікрорівні

Математичні моделі на мікрорівні.

Математичними моделями на мікрорівні є диференціальні рівняння в частинних похідних або інтегральні рівняння, що описують поля фізичних величин. Іншими словами, на мікрорівні використовуються моделі з розподіленими параметрами. У якості незалежних змінних у моделях можуть фігурувати просторові змінні x_1, x_2, x_3 і час t .

Характерними прикладами моделей можуть служити рівняння математичної фізики разом із заданими крайовими умовами.

Наприклад:

1) рівняння теплопровідності

де C — питома теплоємність, ρ — щільність, T — температура, t — час, λ — коефіцієнт теплопровідності, g — кількість теплоти, що виділяється за одиницю часу в одиниці об'єму;

2) рівняння дифузії

де N — концентрація часток, D — коефіцієнт дифузії;

3) рівняння неперервності, що використовуються у фізиці напівпровідникових приладів:

- для дірок
- для електронів
- і рівняння Пуассона

Тут p і n — концентрації дірок і електронів; q — заряд електрона; J_p і J_n щільності дірочного й електронного струмів; g_p і g_n — швидкості процесів генерації-рекомбінації дірок і електронів; E — напруженість електричного поля; ρ — густина електричного заряду; ϵ і ϵ_0 діелектрична проникність і діелектрична постійна.

Крайові умови включають початкові умови, що характеризують просторовий розподіл залежних змінних у початковий момент часу, і граничні, що задають значення цих змінних на границях розглянутої області у функції часу.

Методи аналізу на мікрорівні.

У САПР рішення диференціальних або інтегро-дифференціальних рівнянь з частками похідними виконується чисельними методами. Ці методи засновані на дискретизації незалежних змінних — їхньому представленні кінцевою множиною значень в обраних вузлових точках досліджуваного простору. Ці точки розглядаються як вузли деякої сітки, тому використовувані в САПР методи — *це сіткові методи*.

Серед сіткових методів найбільше поширення одержали два методи:

- метод скінчених різностей (МКР)
- і метод скінчених елементів (МКЕ).

Зазвичай виконують дискретизацію просторових незалежних змінних, тобто використовують просторову сітку. У цьому випадку результатом дискретизації є система звичайних диференціальних рівнянь для нестационарної, або система алгебраїчних рівнянь для стаціонарної задачі.

Нехай необхідно вирішити рівняння

з заданими крайовими умовами

де L і M — оператори диференціювання, $V(z)$ — фазова змінна, $z=(x_1, x_2, x_3)$ — вектор незалежних змінних, $f(z)$ і $\psi(z)$ — задані функції незалежних змінних.

В методі скінчених різниць алгебралізація похідних по просторових координатах базується на апроксимації похідних скінчено-різницевиими виразами. При використанні методу потрібно вибрати кроки сітки по кожній координаті і вид шаблону. Під шаблоном розуміють безліч вузлових точок, у яких значення змінних використовують для апроксимації похідної в одній конкретній точці.

Приклади шаблонів для одновимірних і двовимірних задач приведені на рис. 11. На цьому рисунку кружком більшого діаметра позначені вузли, в яких апроксимується похідна. Чорними крапками позначені вузли, в яких значення фазової змінної входять в апроксимуючий вираз. Число, записане біля вузла, дорівнює коефіцієнтові, з яким значення фазової змінної входить в апроксимуючий вираз. Так, для одновимірних шаблонів

систем

у верхній частині рисунка показана апроксимація похідної $xV\partial\partial$ у точці k , і зазначеним шаблонам при їхньому перегляді зліва направо відповідають апроксимації де h — крок дискретизації по осі x .

Шаблони для двовимірних задач у нижній частині рис. 3.11 відповідають наступним кінцево-різницеvim операторам:

- лівий рисунок -
- середній рисунок -
- правий рисунок -

Тут $V_{k,j}$ — значення V точки (x_{1k}, x_{2j}) ; прийняті однакові значення кроків h по обох координатах.

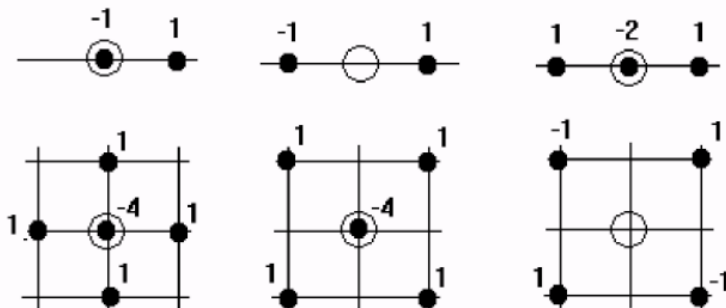


Рис. 11. Приклади шаблонів для метода скінчених різниць.

Метод скінчених елементів базується на апроксимації не похідних, а самого рішення $V(z)$. Але оскільки воно невідоме, то апроксимація виконується виразами з невідомими коефіцієнтами q_i

$$(3.34)$$

де q - вектор-рядок невідомих коефіцієнтів, $\varphi(z)$ — вектор-стовпець координатних (інакше опорних) функцій, заданих так, що задовольняються граничні умови.

При цьому мова йде про апроксимацію рішення в межах скінчених елементів, а з врахуванням їхніх малих розмірів можна говорити про використання порівняно простих апроксимуючих виразів $U(z)$ (напр., $\varphi(z)$ — поліноми низьких ступенів). У результаті підстановки $U(z)$ у вихідне диференціальне рівняння і

виконання операцій диференціювання одержуємо систему невязок

$$(3.35)$$

з якої потрібно знайти вектор Q .

Цю задачу (визначення Q) вирішують одним з наступних методів:

- 1) *метод колокації*, у якому, використовуючи (3.35), формують n рівнянь з невідомим вектором Q : де n — число невідомих коефіцієнтів;
- 2) *метод найменших квадратів*, заснований на мінімізації квадратів невязань (3.35) у n точках або в середньому по області, що розглядається;
- 3) *метод Гальоркіна*, за допомогою якого мінімізуються в середньому по області невязання з ваговими коефіцієнтами, що задаються спеціально.

Найбільше поширення МСЕ одержав у САПР машинобудування для аналізу міцності об'єктів. Для цієї задачі можна використовувати розглянутий підхід, тобто виконати алгебралізацію вихідного рівняння пружності (рівняння Ламе). Однак більш зручним у реалізації МСЕ виявився підхід, заснований на варіаційних принципах механіки.

МКЕ в програмах аналізу механічної міцності.

Як вихідне положення приймають варіаційний принцип Лагранжа (принцип потенційної енергії), відповідно до якого рівноважний стан, у яке може прийти система, характеризується мінімумом потенційної енергії.

192Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Лекція 9. Технічне забезпечення процесу проектування

Потенційна енергія Π визначається як різниця енергії \mathcal{E} деформації тіла і роботи A масових і прикладених поверхневих сил.

У свою чергу

$$(3.36)$$

де σ — вектор-рядок деформацій, $\sigma = (\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23})$ — вектор-стовпець напруг, R — область, яку розглядають. Деформації $i \in \mathcal{R}$ можна виразити через переміщення:

$$(3.37)$$

де W_i — переміщення уздовж осі x_i або в матричній формі:

$$(3.38)$$

де S — очевидний оператор диференціювання з (3.37).

Деформації і напруги зв'язані між собою за допомогою матриці D , що характеризує пружні властивості середовища, яке представлено в табл. 2: (3.39)

Таблиця 2

Коефіцієнти λ і μ , що фігурують у таблиці, називають постійними Ламе, вони виражають пружні властивості матеріалу деталі.

Підставляючи (3.39) і (3.38) у (3.36), одержуємо

Рішенням задачі повинне бути поле переміщень $W(X)$, де $X=(x_1, x_2, x_3)$. Відповідно до МСЕ це рішення апроксимується за допомогою функцій (3.34), які стосовно до сукупності скінчених елементів представимо в матричній формі:

де N — матриця координатних функцій, Q — вектор невідомих коефіцієнтів. Заміняючи $W(X)$ на $U(X)$, одержуємо

$$(3.40)$$

де K — матриця жорсткості.

Відповідно до принципу потенційної енергії в стані рівноваги маємо або, диференціюючи (3.40), знаходимо

$$(3.41)$$

де F — вектор навантажень. Таким чином, задача аналізу міцності, згідно МСЕ, зведена до рішення системи лінійних алгебраїчних рівнянь (3.41).

Матриця жорсткості також виявляється сильно розрідженою, тому для рішення (3.41) застосовують методи розріджених матриць.

Примітка. Одним із широко відомих методів розріджених матриць є метод прогону, застосовуваний у випадку трьохдіагональних матриць коефіцієнтів у системі алгебраїчних рівнянь.

Структура програм аналізу по МСЕ на мікрорівні.

Основними частинами програми аналізу по МСЕ є бібліотеки скінчених елементів, препроцесор, процесор і постпроцесор.

Бібліотеки скінчених елементів (SE) містять моделі елементів — їх матриці твердості. Очевидно, що моделі SE будуть різними для :

- різних задач (аналіз деформацій, полів температур, електричних потенціалів і т.п.),
- різних форм SE (наприклад, у двовимірному випадку — трикутні або чотирикутні елементи),
- різних наборів координатних функцій.

Вихідні дані для *препроцесора* — геометрична модель об'єкта, найчастіше одержувана з підсистеми конструювання. Основна функція препроцесора — представлення досліджуваного середовища (деталі) у сітковому виді, тобто у виді множини скінчених елементів.

Процесор — програма, що асемблює (збирає) моделі окремих SE в загальну систему алгебраїчних рівнянь (3.41) і розв'язує цю систему одним з методів розріджених матриць.

Постпроцесор служить для візуалізації результатів рішення в зручній для користувача формі. У машинобудівних САПР це графічна форма. Користувач може бачити початкову (до навантаження) і

систем

деформовану форми деталі, поля напруг, температур, потенціалів і т.п. у виді кольорових зображень, у яких палітра кольорів або інтенсивність світіння характеризують значення фазової змінної.

Світовими лідерами серед програм скінчено-елементного аналізу механічних деталей є програмно-методичні комплекси Nastran, Ansys, Nisa, Adina, Cosmos.

Як правило, ці комплекси містять у собі ряд програм, споріднених по математичному забезпеченню, інтерфейсам, та деяким використовуваним модулям. Ці програми розрізняються орієнтацією на різні додатки, ступенем спеціалізації, ціною або виконуваною обслуговуючою функцією. Наприклад, у комплексі Ansys основні модулі дозволяють виконувати аналіз механічної міцності, теплопровідності, динаміки рідин і газів, акустичних і електромагнітних полів. В усі варіанти програм входять пре- і постпроцесори, а також інтерфейс із базою даних. Передбачено експорт (імпорт) даних між Ansys і ведучими комплексами геометричного моделювання і машинної графіки.

17. ММ аналізу на функціонально-логічному рівні

Моделювання й аналіз аналогових пристроїв.

На функціонально-логічному рівні досліджують пристрої, елементами яких є досить складні вузли і блоки, які були *системами на макрорівні*. Тому необхідно спростити представлення моделей цих вузлів і блоків у порівнянні з їхнім представленням на макрорівні. Іншими словами, замість повних моделей вузлів і блоків потрібно використовувати їхні макромоделі.

Замість двох типів фазових змінних у моделях функціонально-логічного рівня фігурують змінні одного типу, які називають *сигналами*. Фізичний зміст сигналу, тобто його віднесення до фазових змінних типу потоку або типу потенціалу, конкретизують у кожному конкретному випадку, виходячи з особливостей задачі.

Основою моделювання аналогових пристроїв на функціонально-логічному рівні є використання апарата передатних функцій. При цьому, модель кожного елемента представляють у виді рівняння «вхід-вихід», тобто у виді

$$V_{\text{ВЫХ}} = f(V_{\text{ВХ}}), \quad (3.42)$$

де $V_{\text{ВЫХ}}$ і $V_{\text{ВХ}}$ — сигнали на виході і вході вузла, відповідно. Якщо вузол має більш ніж один вхід і один вихід, то в цьому рівнянні скаляри $V_{\text{ВЫХ}}$ і $V_{\text{ВХ}}$ стають векторами.

Однак відомо, що представлення моделі у виді (3.42) можливо тільки тоді, якщо вузол є безінерційним, тобто в повній моделі вузла не фігурують похідні. Отже, для одержання (3.42) у загальному випадку потрібно попередня *алгебралізація повної моделі*. Таку алгебралізацію виконують за допомогою інтегральних перетворень, наприклад, за допомогою перетворення Лапласа, переходячи з часової області в простір комплексної зінної p . Тоді в моделях типу (3.42) мають місце не оригінали, а зображення сигналів $V_{\text{ВЫХ}}(p)$ і $V_{\text{ВХ}}(p)$. А самі моделі реальних блоків намагаються, по можливості, максимально спростити і представити їх моделями типових блоків (ланок) з числа заздалегідь розроблених бібліотечних моделей. Зазвичай, моделі ланок мають вигляд:

$$V_{\text{ВЫХ}}(p) = h(p)V_{\text{ВХ}}(p),$$

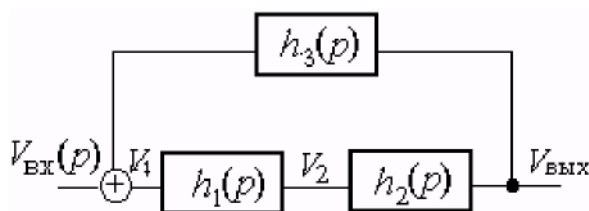
де $h(p)$ — передатна функція ланки.

У випадку застосування перетворення Лапласа з'являються обмеження на використання нелінійних моделей, а саме: у моделях не повинне бути нелінійних інерційних елементів.

Інше допущення, що спрощує моделювання на функціонально-логічному рівні — неврахування впливу навантаження на характеристики блоків. І дійсно, підключення до виходу блоку деякого іншого вузла ніяк не впливає на модель блоку (3.42).

Власне одержання ММС (системи) із ММЕ (елементів) виявляється, за рахунок прийнятих допущень, значно простішим, ніж на макрорівні: *ММС є сукупність ММЕ, у яких ототожнені сигнали на з'єднаних входах і виходах елементів*. Ця ММС являє собою систему алгебраїчних рівнянь.

Одержання ММС проілюструємо простим прикладом (рис. 3.12), де показано систему з трьох блоків з передатними функціями $h_1(p)$, $h_2(p)$ і $h_3(p)$. ММС має вигляд:



$$\begin{aligned} V_2 &= h_1(p)V_1; \\ V_{\text{ВЫХ}}(p) &= h_2(p)V_2; \\ V_1 &= V_{\text{ВХ}}(p) + h_3(p)V_{\text{ВЫХ}}(p) \end{aligned}$$

або

$$V_{\text{ВЫХ}}(p) = H(p)V_{\text{ВХ}}(p),$$

Рис. 3.12. Приклад схеми з трьох блоків

де

$$H(p) = h_1(p) h_2(p) / (1 - h_1(p) h_2(p) h_3(p))$$

Таким чином, аналіз зводиться до наступних операцій:

1. задану схему пристрою представляють сукупністю ланок і, якщо схема не повністю покривається типовими ланками, то розробляють оригінальні моделі;
2. формують ММС із моделей ланок;
3. застосовують пряме перетворення Лапласа до вхідних сигналів;
4. розв'язують систему рівнянь ММС і знаходять зображення вихідних сигналів;
5. за допомогою зворотнього перетворення Лапласа повертаються в часову область з області комплексної змінної.

Математичні моделі дискретних пристроїв.

Аналіз дискретних пристроїв на функціонально-логічному рівні потрібний, насамперед, при проектуванні пристроїв обчислювальної техніки і цифрової автоматики. Тут додатково до допущень, прийнятих при аналізі аналогових пристроїв, використовують дискретизацію сигналів, причому базовим є двозначне представлення сигналів – "істина" (1) і "неправда" (0), а самі сигнали розглядають як булеві величини. Тоді, для моделювання можна використовувати апарат математичної логіки. Можуть застосуватись також трьох- і більш- значні моделі. Зміст значень сигналів у багатозначному моделюванні і причини його застосування будуть пояснені нижче на деяких прикладах.

Елементами цифрових пристроїв на функціонально-логічному рівні служать елементи, що виконують логічні функції і можливо функції збереження інформації. Найпростішими елементами є диз'юнктор, кон'юнктор, інвертор, що реалізують відповідно операції диз'юнкції (АБО) $y = a \text{ or } b$, кон'юнкції (І) $y = a \text{ and } b$, заперечення (НЕ) $y = \text{not } a$, де y - вихідний сигнал, a і b — вхідні сигнали. Число входів може бути і більш двох. Умовні схемні позначення простих логічних елементів показані на рис. 3.13.

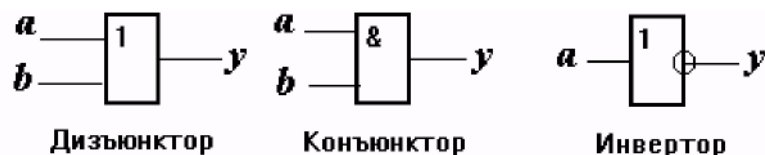


Рис 3.13. Умовні позначки логічних елементів на схемах

Математичні моделі пристроїв являють собою систему математичних моделей елементів, що входять у пристрій, при ототожненні сигналів, що відносяться до того самого з'єднання елементів.

Лекція 9. Технічне забезпечення процесу проектування

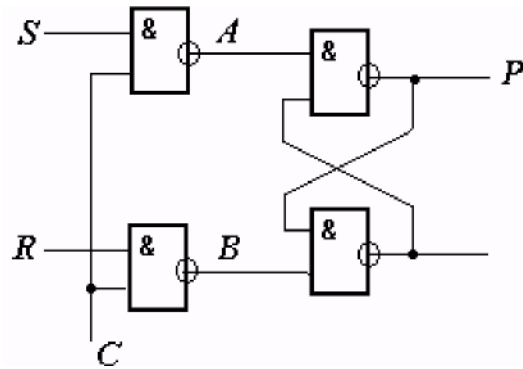


Рис 3.14. Схема RS-тригера

Розрізняють *синхронні* й *асинхронні* моделі.

Синхронна модель являє собою систему логічних рівнянь, у якій відсутній така змінна як час. Синхронні моделі використовують для аналізу сталих станів.

Прикладом синхронної моделі може служити наступна система рівнянь, отримана для логічної схеми тригера (рис. 3.14):

$$B = \text{not } (R \text{ and } C); Q = \text{not } (B \text{ and } P); P = \text{not } (A \text{ and } Q);$$

$$A = \text{not } (S \text{ and } C).$$

Асинхронні моделі відбивають не тільки логічні функції, але і часові затримки в поширенні сигналів. Асинхронна модель логічного елемента має вигляд

$$Y(t+t_{3d}) = f(X(t)), \quad (3.43)$$

де t_{3d} - затримка сигналу в елементі; f - логічна функція. Запис (3.43) означає, що вихідний сигнал y приймає значення логічної функції, що відповідає значенням аргументів $X(t)$, у момент часу $t+t_{3d}$. Отже, асинхронні моделі можна використовувати для аналізу динамічних процесів у логічних схемах.

Терміни синхронна й асинхронна моделі можна пояснити орієнтованістю цих моделей на синхронні й асинхронні схеми, відповідно. У синхронних схемах передача сигналів між цифровими блоками відбувається тільки при подачі на спеціальні синхро-входи тактових (синхронізуючих) імпульсів. Частота тактових імпульсів вибирається такою, щоб до моменту приходу синхроімпульсу перехідні процеси від попередніх передач сигналів фактично закінчилися. Отже, у синхронних схемах розрахунок затримок не актуальний, швидкодія пристрою визначається завданням тактової частоти.

Синхронні моделі можна використовувати не тільки для виявлення принципових помилок у схемній реалізації заданих функцій. З їхньою допомогою можна виявляти місця в схемах, небезпечні, з погляду, виникнення в них спотворюючих завад. Ситуації, зв'язані з потенційною небезпекою виникнення завад і збоїв, називають *ризиками збою*.

Розрізняють статичний і динамічний ризики збою.

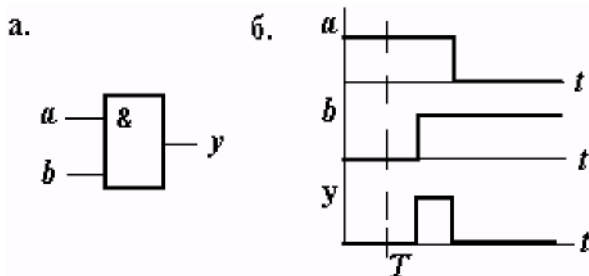


Рис. 3.15. Статичний ризик збою: а - схема; б - діаграма сигналів

Статичний ризик збою ілюструє ситуація рис. 3.15, якщо на два входи елемента И можуть приходити перепади сигналів у протилежних напрямках, як це показано на рис. 3.15,б. Якщо замість ідеального випадку, коли обидва перепади приходять у момент часу T , перепади внаслідок розкиду затримок прийдуть неодноразомно, причому так, як показано на рис. 3.15,б, то на

СИСТЕМ

виході елемента з'являється імпульс перешкоди, що може спотворити роботу всього пристрою. Для усунення таких ризиків збою потрібно вміти їх виявляти. З цією метою застосовують тризначне синхронне моделювання.

При цьому трьома можливими значеннями сигналів є 0, 1 і ⊗, причому значення ⊗ інтерпретується як невизначеність. Правила виконання логічних операцій І, АБО, НЕ в тризначному алфавіті очевидні з розгляду табл. 3.6. У ній другий рядок відведений для значень одного аргументу, а перший стовпець — для значень другого аргументу, значення функцій представлені нижче другого рядка і правіше першого стовпця.

	Операція			Значення	a	y	y
	И	ИЛИ	НЕ				
	0 ⊗ 1	0 ⊗ 1	0 ⊗ 1	вихідні	1	0	0
0	0 0 0	0 ⊗ 1	1 ⊗ 0	проміжні	<8	<	<8>
⊗	0 ⊗ ⊗	⊗ ⊗ 1	-				
1	0 ⊗ 1	1 1 1	-	підсумкові	0	1	0

Таблиця 3.6.

Таблиця 3.7

При аналізі ризиків збою на кожному такті замість однократного рішення рівнянь моделі проводять дворазове рішення. Тому, можна говорити про початкові, проміжні (після першого рішення) і результуючі (після другого рішення) значеннях змінних. Для вхідних сигналів припустимі тільки такі послідовності початкових, проміжних і результуючих значень: 0-0-0, 1-1-1, 0-⊗-1, 1-⊗-0. Для інших змінних поява послідовності 0-⊗-0 або 1-⊗-1 означає невизначеність під час перехідного процесу, тобто можливість статичного ризику збою.

Для найпростішої схеми (рис. 3.15,а) результати тризначного моделювання представлені в табл. 3.7.

Динамічний ризик збою ілюструють схема і тимчасові діаграми рис. 3.16. Збій виражається в появі замість одного перепаду на виході, що має місце при правильному функціонуванні, декількох перепадів. Виявлення динамічних ризиків збою також виконують за допомогою дворазового рішення рівнянь моделі, але при використанні п'ятизначного алфавіту з безліччю значень (0, 1, ⊗, β), де α інтерпретується як позитивний перепад, β — як негативний перепад, інші символи мають колишній сенс. Під час відсутності збоїв послідовності значень змінних у вихідному, проміжному і підсумковому станах можуть бути такими: 0-0-0, 1-1-1, 0-α-1, 1-β-0. Послідовності 0-⊗-1 або 1-⊗-0 вказують на динамічний ризик збою.

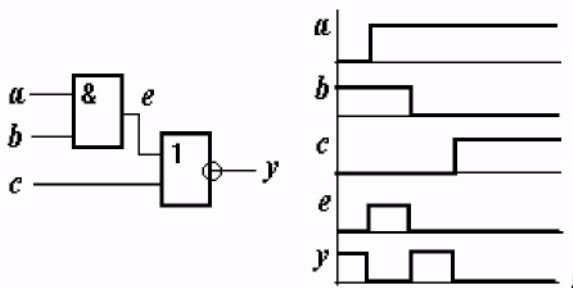


Рис. 3.16. Динамічний ризик збою

Тризначний алфавіт можна використовувати й в асинхронних моделях.

Нехай у моделі $y(t+t_i)=Y(X)$ у момент часу t_i входи X такі, що в момент часу $y(t+t_i)$ відбувається переключення вихідного сигналу. Але якщо враховувати розкид затримок, то прийматиме деяке випадкове значення в діапазоні _____ отже, у моделі в інтервалі часу від _____ сигнал у повинний мати невизначене значення Д. Саме це і досягається за допомогою тризначного асинхронного моделювання.

Методи логічного моделювання.

У відношенні асинхронних моделей можливі два методи моделювання: (1) покроковий (інкрементний) і (2) подійний.

1) У покроковому методі час дискретизується й обчислення по моделі виконуються в 198Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Лекція 9. Технічне забезпечення процесу проектування

дискретні моменти часу t_0, t_b, t_2, \dots і т.д. Крок дискретизації обмежений зверху значенням допустимої похибки визначення затримок виявляється досить малим, а час аналізу значним.

2) Для скорочення часу аналізу використовують **подійний метод**. У цьому методі подією називають зміну будь-якої змінної моделі. Подійне моделювання засноване на наступному правилі: *звертання до моделі логічного елемента відбувається тільки в тому випадку, якщо на входах цього елемента відбулася подія.*

Тому, оскільки у складних логічних схемах на кожному такті синхронізації звичайно відбувається переключення усього лише 2-3% логічних елементів і, відповідно, у подійному методі в кілька разів зменшуються обчислювальні витрати в порівнянні з покроковим моделюванням.

Методи аналізу синхронних моделей являють собою методи рішення систем логічних рівнянь. До цих методів відносяться **(1) метод простих ітерацій** і **(2) метод Зейделя**, що аналогічні однойменним методам рішення систем алгебраїчних рівнянь у неперервній математиці.

Застосування цих методів до моделювання логічних схем зручно проілюструвати на прикладі схеми тригера (див. рис. 3.14).

Таблиця 3.8

Ітерація							
Попередній стан							
Вихідні значення (ітерація 0)							
Ітерація 1							
Ітерація 2				*			*
Ітерація 3						*	
Ітерація 4						*	

У табл. 3.8 значення змінної моделі в вихідному стані і після кожної ітерації відповідно до методу простих ітерацій. У вихідному стані задають початкові (можливо довільні) значення проміжних і вихідних змінних, у даному прикладі це значення змінних Y, Q, P, A , що відповідають попередньому станові тригера. Але стан тригера повинен відповідати зазначеним у таблиці змінившимся значенням вхідних сигналів R, S і C . Обчислення закінчуються, якщо на черговій ітерації змін змінних ні, що і спостерігається в даному прикладі на четвертій ітерації.

Відповідно до методу простих ітерацій, у праві частини рівнянь моделі на кожній ітерації підставляють значення змінних, отримані на попередній ітерації. На відміну від цього в методі Зейделя, якщо в деякої змінної обновилося значення на поточній ітерації, то саме його і використовують у подальших обчисленнях уже на поточній ітерації. Метод Зейделя дозволяє скоротити число ітерацій, але для цього потрібно попередньо упорядкувати рівняння моделі так, щоб послідовність обчислень відповідала послідовності проходження сигналів за схемою. Таке упорядкування виконують за допомогою *ранжирування*.

Ранжирування полягає в присвоєнні елементам і змінним моделі значень рангів у відповідності з наступними правилами: 1) у схемі розриваються всі контури зворотного зв'язку, що приводить до появи додаткових входів схеми (псевдовходів); 2) усі зовнішні змінні (у тому числі на псевдовходах) одержують ранг 0; 3) елемент і його вихідні змінні одержують ранг k , якщо в елемента усі входи проранжировані і старший серед рангів входів дорівнює $k-1$.

Так, якщо в схемі (див. рис. 3.14) розірвати наявний контур зворотного зв'язку в ланцюзі змінної Q і позначити змінну на псевдовході Q_b то ранги змінних виявляються наступними: R, S, Z, Q_1 мають ранг 0, A, B — ранг 1, P — ранг 2 і Q — ранг 3. Відповідно до цього переупорядковують рівняння

систем

в моделі тригера:

$$A = \text{not } (S \text{ and } C). B = \text{not } (R \text{ and } C); P = \text{not } (A \text{ and } Q); Q = \text{not } (B \text{ and } P).$$

Тепер уже на першій ітерації по Зейделю одержуємо необхідний результат. Якщо розірвати контур зворотного зв'язку в ланцюзі змінної Р, то рішення в даному прикладі буде отримано після другої ітерації, але це все рівно помітно швидше, ніж при використанні методу простої ітерації.

приклад

Для тригера (рис. 3.14) застосування подійного моделювання в рамках метода простих ітерацій призводить до зменшення обсягу розрахунків: замість 16-кратних звертань до моделей елементів (як це видно з табл. 3.8), виконується лише 5-кратне звернення. В табл. 3.8 зірочками відмічені значення змінних, які обчислені в подійному методі.

Так, на ітерації 0 мають місце зміни змінних S і C, тому на наступній ітерації звертання відбувається тільки до моделей елементів з виходами А і В.

18. Математичне забезпечення СОС на системному рівні

Основні поняття з теорії масового обслуговування (ТМО).

Об'єктами проектування на системному рівні є такі складні системи, як виробничі підприємства, транспортні системи, обчислювальні системи і мережі, автоматизовані системи проектування і керування і т.п.

У цих системах аналіз процесів функціонування систем зв'язаний з дослідженням проходження через *систему потоку заявок* (їх ще називають *вимогами* або *транзактами*). Розроблювачів подібних складних систем цікавлять насамперед такі параметри, як

- продуктивність (пропускна здатність) проектованої системи,
- тривалість обслуговування (затримки) заявок у системі,
- ефективність використовуваного в системі устаткування.

Заявками можуть бути замовлення на виробництво виробів, задачі, розв'язувані в обчислювальній системі, клієнти в банках, вантажі, що надходять на транспортування й ін. Очевидно, що параметри заявок, що надходять у систему, є випадковими величинами і при проектуванні можуть бути відомі лише їхні закони розподілу і числові характеристики цих розподілів.

Тому, аналіз функціонування на системному рівні, як правило, носить статистичний характер. Як математичний апарат моделювання зручно прийняти **теорію масового обслуговування (ТМО)**, а як моделі систем на цьому рівні використовувати **системи масового обслуговування (СМО)**.

Типовими вихідними параметрами в СМО є числові характеристики таких величин, як:

- час обслуговування заявок у системі,
- довжини черг заявок на входах,
- час чекання обслуговування в чергах,
- завантаження пристроїв системи,
- ймовірність обслуговування в заданий термін,
- тощо.

У найпростішому випадку СМО являє собою деякий пристрій, що називають *обслуговуючим апаратом (ОА)*, разом з чергами заявок на входах. Більш складні СМО складаються з багатьох взаємозалежних ОА. Обслуговуючі апарати СМО в сукупності утворюють статичні об'єкти СМО, які ще називають ресурсами. Напр., в обчислювальних мережах ресурси представлені апаратними і програмними засобами.

У СМО, крім статичних об'єктів, фігурують динамічні об'єкти — *транзакти*. Напр., в обчислювальних мережах динамічними об'єктами є задачі, що розв'язуються, і запити на інформаційні послуги.

Стан СМО характеризується станами складових її об'єктів. Напр., стани ОА виражаються булевими величинами, значення яких інтерпретуються як *true (зайнято)* і *false (вільно)*, і довжинами черг на входах ОА, що приймають позитивні цілочисленні значення. Змінні, що характеризують стан СМО, називають *змінними станами* або *фазовими змінними*.

Правило, відповідно до якого заявки вибирають з черг на обслуговування, називають *дисципліною обслуговування*, а величину, що виражає переважне право на обслуговування, — пріоритетом. У *безпріоритетних дисциплінах* усі транзакти мають однакові пріоритети. Серед безпріоритетних дисциплін найбільш популярні дисципліни:

1. FIFO (1-й прийшов – 1-й обслуговується),
2. LIFO (останній прийшов — 1-й обслуговується) і
3. з випадковим вибором заявок з черг.

У пріоритетних дисциплінах для заявок кожного пріоритету на вході ОА виділяється своя

систем

черга. Заявка з черги з низьким пріоритетом надходить на обслуговування, якщо порожні черги з більш високими пріоритетами. Розрізняють пріоритети *абсолютні, відносні і динамічні*. Заявка з черги з більш високим *абсолютним пріоритетом*, надходячи на вхід зайнятого ОА, перериває вже почате обслуговування заявки більш низького пріоритету. У випадку відносного пріоритету переривання не відбувається, більш високопріоритетна заявка чекає закінчення вже початого обслуговування. Динамічні пріоритети можуть змінюватися під час перебування заявки в СМО.

Дослідження поведінки СМО, тобто визначення тимчасових залежностей змінних, що характеризують стан СМО, при подачі на входи деяких експериментальних потоків заявок, називають ***імітаційним моделюванням СМО***. Імітаційне моделювання проводять шляхом відтворення подій, що відбуваються одночасно або послідовно в модельному часі. При цьому, під подією розуміють факт зміни значення будь-якої фазової змінної.

Підхід, альтернативний імітаційному, називають *аналітичним дослідженням СМО*. Аналітичне дослідження полягає в одержанні формул для розрахунку вихідних параметрів СМО з наступною підстановкою значень аргументів у ці формули в кожному конкретному експерименті.

Моделі СМО, які використовують при імітаційному й аналітичному моделюванні, називають відповідно ***імітаційними й аналітичними***.

Аналітичні моделі зручні у використанні, оскільки для аналітичного моделювання не потрібні великі витрати обчислювальних ресурсів. Часто без постановки спеціальних обчислювальних експериментів розроблювач може оцінити характер впливу аргументів на вихідні параметри, виявити загальні закономірності в поведінці системи. Але, на жаль, аналітичне дослідження вдається реалізувати тільки для окремих випадків - порівняно нескладних СМО. Для складних СМО аналітичні моделі вдається одержати тільки при прийнятті допущень, що значно спрощують систему і ставлять під сумнів адекватність моделі.

Тому, основним підходом до аналізу СОС на системному рівні проектування вважають імітаційне моделювання, а аналітичне дослідження використовують при попередній оцінці різних пропонувананих варіантів систем.

Деякі компоненти СМО характеризуються більш ніж одним вхідним і (або) вихідним потоками заявок. Правила вибору одного з можливих напрямків руху заявок входять у відповідні моделі компонентів. В одних випадках такі правила відносяться до вихідним даних (наприклад, вибір напрямку по ймовірності), але в інших випадках бажано знайти оптимальне керування потоками у вузлах розгалуження. Тоді задача моделювання стає більш складною задачею синтезу. Характерними прикладами є маршрутизація заявок або синтез розкладів і планів.

Аналітичні моделі СМО.

Як відзначено вище, аналітичні моделі СМО вдається одержати лише при досить серйозних допущеннях. До числа типових допущень відносяться наступні:

По-перше, вважають, що в СМО використовуються безпріоритетні дисципліни обслуговування типу FIFO.

По-друге, часи обслуговування заявок у пристроях вибираються відповідно до експонентного закону розподілу.

По-третє, в аналітичних моделях СМО вхідні потоки заявок апроксимуються найпростішими потоками, тобто потоками, що володіють властивостями стаціонарності, ординарності

202 Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Лекція 9. Технічне забезпечення процесу проектування

(неможливості одночасного надходження двох заявок на вхід СМО), відсутності післядії.

У більшості випадків моделі СМО відображають процеси з кінцевою безліччю станів і з відсутністю післядії. Такі процеси називають кінцевими марковськими ланцюгами.

Марковські ланцюги характеризуються безліччю станів S , матрицею імовірностей переходів з одного стану в інше і початковими умовами (початковим станом). Зручно представляти марковський ланцюг у виді графа, у якому:

- вершини відповідають станам ланцюга,
- дуги — переходам,
- ваги дуг — імовірностям переходів (якщо час дискретний) або інтенсивностям переходів (якщо час неперервний).

Інтенсивністю переходу називають величину $V_{ij} = \lim P_{ij}(t_1)/t_1$, при $t_1 \rightarrow 0$, де $P_{ij}(t_1)$ — імовірність переходу зі стану S_i у стан S_j за час t_1 .

Зазвичай приймається умова: $V_{ij} = -\sum_{j \neq i} V_{ij}$, що означає $\sum_{j=1}^N V_{ij} = 0$, де N — число станів.

На рис. 3.17 наведено приклад марківського ланцюга у виді графа зі станами S_1, \dots, S_4 , а в табл. 3.9 представлена матриця інтенсивностей переходів для цього прикладу.

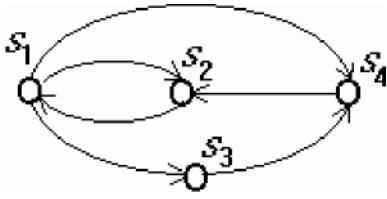


Рис.3.17. Приклад марківського ланцюга

тан	S_1	2	3	4
1	$-V_{12}$ $V_{13}-V_{14}$	12	13	14
2	V_{21}	V_{21}		
3	0		V_{34}	34
4	0	42		V_{42}

Таблиця 3.9

Більшість вихідних параметрів СМО можна визначити, використовуючи інформацію про поведінку СМО, тобто інформацію про стани СМО в сталих (стаціонарних) режимах і про їх зміни в перехідних процесах. Ця інформація має ймовірнісну природу, що обумовлює опис поведінки СМО в термінах ймовірностей перебування системи в різних станах. Основою такого опису, а отже, і багатьох аналітичних моделей СМО є **рівняння Колмогорова**:

$$dP_i/dt = \sum_j (V_{ji} P_j) - P_i \sum_k V_{ik}.$$

Де $P_i(t)$ і $P_j(t)$ — ймовірності перебування системи в станах S_i і S_j , відповідно, в момент часу t , $V_{ji}(t_1)$, $V_{ik}(t_1)$ — ймовірності зміни станів протягом часу t_1 ; j, k — множини індексів інцидентних вершин стосовно вершини S_i по вхідних і вихідних дугах на графі станів, відповідно.

У стаціонарному стані $dP_i/dt = 0$ і рівняння Колмогорова складають систему алгебраїчних рівнянь, у якій i -й вузол представлений рівнянням: $\sum_j (V_{ji} P_j) = P_i \sum_k V_{ik}$. (3.46)

систем

Додаючи $V_{ii}P_i$ до лівої і правої частин рівняння (3.46) і з огляду на (3.44), одержуємо:

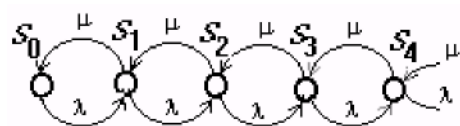
$$\sum_{j=1}^N (V_{ji}P_j) = P_i \sum_{k=1}^N V_{ik} = 0, \quad \text{тобто} \quad \sum_{j=1}^N (V_{ji}P_j) = 0,$$

де P_j — фінальні ймовірності.

Приклад аналітичної моделі.

Прикладом СМО, до якої можна застосувати аналітичні методи дослідження, є одноканальна СМО з найпростішим вхідним потоком інтенсивністю λ і тривалістю обслуговування, що підкоряється експонентному закону обслуговування інтенсивністю μ . Для цієї СМО потрібно одержати аналітичні залежності середнього числа N_{av} заявок, що знаходяться в системі, середню довжину Q_{av} черги до ОА, час T_{av} перебування заявки в системі, час T_{or} очікування в черзі.

На рис. 3.18 представлено граф станів розглянутої СМО, де S_k — стан з k заявками в системі. Матриця інтенсивностей представлена в табл. 3.10.



тан	0	1	2	3	4
0	λ				
1		$\lambda - \mu$			
2			$\lambda - \mu$		
3				$\lambda - \mu$	
4					$\lambda - \mu$

Рис.3.18. Граф станів

Таблиця 3.10

Рівняння Колмогорова для сталого режиму мають вигляд:

$$\begin{aligned} \lambda P_0 + \mu P_1 &= 0, \\ \lambda P_0 - (1+\mu)P_1 + \mu P_2 &= 0, \\ \mu P_1 - (1+\mu)P_2 + \lambda P_3 &= 0, \\ \mu P_2 - (1+\mu)P_3 + \lambda P_4 &= 0, \\ \dots \end{aligned}$$

Використовуючи рівняння Колмогорова, можна виразити всі $P_i, i = 1,2,3,\dots$, через P_0 . Одержимо:

$$\begin{aligned} P_1 &= \lambda P_0 / \mu = a P_0, \\ P_2 &= ((1+\mu)P_1 - \lambda P_0) / \mu = (1+a)P_1 - a P_0 = a^2 P_0; \\ P_3 &= (1+a)P_2 - a P_1 = a^2 P_1 = a^3 P_0 \text{ и т.д.} \end{aligned}$$

Лекція 9. Технічне забезпечення процесу проектування

Тут введено позначення $a = \lambda/\mu$. Відзначимо, що сталий режим можливий тільки при $a < 1$.

Оскільки $\sum_{i=0}^{\infty} P_i = 1$, тому $P_0 = 1 - \sum_{i=1}^{\infty} P_i = 1 - P_0(a + a^2 + a^3 + \dots) = 1 / (1 + a + a^2 + a^3 + \dots) = 1 - a$.

Тепер неважко одержати й інші необхідні результати:

$$N_{av} = \sum_{k=1}^{\infty} P_k k = P_1 + 2P_2 + 3P_3 + \dots = a(1-a)(1 + 2a + 3a^2 + \dots) = a(1-a) / (1-a)^2 = a / (1-a).$$

$$Q_{av} = P_2 + 2P_3 + 3P_4 + \dots = \sum_{k=2}^{\infty} (k-1)P_k = P_0 a^2 (1 + 2a + 3a^2 + \dots) = a^2 / (1-a).$$

Часи перебування в системі і черзі знаходяться із співвідношень:

$$N_{av} = \lambda T_{av}, \quad T_{av} = a / (1-a) / \lambda = 1 / (\mu - \lambda),$$
$$Q_{av} = \lambda T_{or}, \quad T_{or} = a^2 / (1-a) / \lambda = a / (\mu - \lambda).$$

1.3.1.1. Імітаційне моделювання СМО.

Для представлення імітаційних моделей можна використовувати мови програмування загального застосування, однак такі представлення виявляються досить громіздкими. Тому звичайно застосовують спеціальні мови імітаційного моделювання на системному рівні. Серед мов імітаційного моделювання розрізняють мови, орієнтовані на опис подій, засобів обслуговування або маршрутів руху заявок (процесів).

Вибір мови моделювання визначає структуру моделі і методику її побудови.

Для опису імітаційних моделей на системному рівні (такі моделі іноді називають *мережними імітаційними моделями* — МІМ) частіше використовують мови, орієнтовані **на події** (наприклад, мови Симскрипт, SMPL і ряд інших) або **на процеси** (наприклад, мови Симула, SOL, GPSS).

Мови імітаційного моделювання реалізуються в програмно-методичних комплексах моделювання СМО, що мають певний ступінь спеціалізації. Так, комплекси на базі мови GPSS можна використовувати в багатьох додатках, але є спеціалізовані комплекси для моделювання обчислювальних мереж, систем керування підприємствами і т.п.

При використанні мов, орієнтованих на процеси, у складі МІМ виділяються елементарні частини і ними можуть бути джерела вхідних потоків заявок, пристрої, нагромаджувачі і вузли. *Джерело вхідного потоку заявок* являє собою алгоритм, відповідно до якого обчислюються моменти t_k появи заявок на виході джерела. Джерела можуть бути залежними і незалежними. У залежних джерелах моменти появи заявок зв'язані з настанням визначених подій, наприклад, із приходом іншої заявки на вхід деякого пристрою. Типовим незалежним джерелом є алгоритм вироблення значень t_k випадкової величини з заданим законом розподілу.

Пристрої в імітаційній моделі представлені алгоритмами вироблення значень інтервалів (тривалості) обслуговування. Найчастіше це алгоритми генерації значень випадкових величин із заданим законом розподілу. Але можуть бути пристрої з детермінованим часом обслуговування або часом, обумовленим подіями в інших частинах МІМ. Модель пристрою відображає також задану дисципліну обслуговування, оскільки в модель входить алгоритм, що керує чергами на входах пристрою.

Нагромаджувачі моделюються алгоритмами визначення обсягів пам'яті, займаних заявками, що приходять на вхід нагромаджувача. Звичайно обсяг пам'яті, займаний заявкою, обчислюється як значення випадкової величини, закон і (або) числові характеристики розподілу може залежати від типу заявки.

Вузли виконують сполучні, керуючі і допоміжні функції в імітаційній моделі, наприклад, для вибору напрямків руху заявок у ЦІМ, зміни їхніх параметрів і пріоритету, поділу заявок на частини, їх

систем

об'єднання і т.п.

Зазвичай, кожному типові елементарної моделі, за винятком лише деяких вузлів, у програмній системі відповідає визначена процедура (підпрограма). Тоді МІМ можна представити як алгоритм, що складається з упорядкованих звертань до цих процедур, що відбивають поведінку модельованої системи.

У процесі моделювання відбуваються зміни модельного часу, що найчастіше приймається дискретним, вимірюванням у тактах. Час змінюється після того, як закінчена імітація чергової групи подій, що відносяться до сучасного моменту часу t_k . Імітація супроводжується нагромадженням в окремому файлі статистики таких даних, як кількості заявок, що вийшли із системи обслуженими і необслуженими, сумарний час зайнятого стану для кожного з пристроїв, середні довжини черг і т.п. Імітація закінчується тоді, коли поточний час перевищить заданий відрізок часу або коли вхідні джерела вироблять задане число заявок. Після цього проводять обробку накопичених у файлі статистики даних, що дозволяє одержати значення необхідних вихідних параметрів.

Подійний метод моделювання.

У програмах імітаційного моделювання СМО переважно реалізують *подійний метод* організації обчислень. Сутність подійного методу полягає у відстеженні на моделі послідовності подій у тому порядку, у якому вони відбувалися б у реальній системі. Обчислення виконують тільки для тих моментів часу і тих частин (процедур) моделі, до яких відносяться здійснені події. Іншими словами, звертання на черговому такті моделюючого часу здійснюються тільки до моделей тих елементів (пристроїв, нагромаджувачів), на входах яких у цьому такті відбулися зміни. Оскільки зміни станів у кожному такті, зазвичай, спостерігаються лише в малої частки ОА, подійний метод може істотно прискорити моделювання в порівнянні з інкрементним методом, у якому на кожному такті аналізуються стани всіх елементів моделі.

Розглянемо можливу схему реалізації подійного методу імітаційного моделювання.

Моделювання починається з перегляду операторів генерування заявок, тобто зі звертання до моделей джерел вхідних потоків. Для кожного незалежного джерела таке звертання дозволяє розрахувати момент генерації першої заявки. Цей момент разом з ім'ям — посиланням на заявку — заноситься в список майбутніх подій (СМП), а повідомлення про генеровану заявку — у список заявок (СЗ). Запис у СЗ містить у собі ім'я заявки, значення її параметрів (атрибутів), місце, яке вона займає в даний момент у МІМ. У СМП події упорядковуються по збільшенню моментів настання.

Потім зі СМП вибирають сукупність повідомлень про події, що відносяться до найбільш раннього моменту часу. Ця сукупність переноситься в список поточних подій (СПП), з якого витягаються посилання на події. Звертання по посиланню до СЗ дозволяє установити місце заявки А в МІМ, з яким пов'язана модельована подія. Нехай цим місцем є пристрій Х. Далі програма моделювання виконує наступні дії (рис. 3.19):

1. змінює параметри стану пристрою Х; наприклад, якщо заявка А звільняє Х, а черга до Х не була порожня, то відповідно до заданої дисципліни про обслуговування з черги до Х вибирається заявка В і надходить на обслуговування в Х;
2. прогнозується час настання наступної події, пов'язаної з заявкою В, шляхом звертання до моделі пристрою Х, у якій розраховується тривалість обслуговування заявки В; повідомлення про цю майбутню подію заносяться в СМП і СЗ;
3. відбувається імітація руху заявки А в МІМ по маршруту, обумовленому заданою програмою моделювання до тих пір, поки заявка не прийде на вхід деякого ОА; тут заявка або затримується

206 Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Лекція 9. Технічне забезпечення процесу проектування

в черзі, або шляхом звертання до моделі цього ОА прогнозується настання майбутньої події, пов'язаної з подальшою долею заявки A ; повідомлення про цю майбутню подію також заносяться в СМП і СЗ;

- у файл статистики додаються необхідні дані.

Після обробки всіх подій, що відносяться до моменту часу t_k відбувається збільшення модельного часу до значення, що відповідає найближчій майбутній події, і розглянутий процес імітації повторюється.

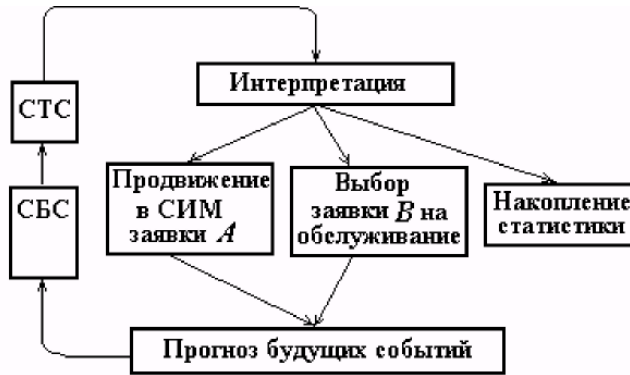


Рис. 3.19. Ілюстрація подійного моделювання

1.3.1.2. Мережі Петрі.

Мережі Петрі — це апарат для моделювання динамічних дискретних систем (переважно асинхронних рівнобіжних процесів). Мережа Петрі визначається як четвірка $\langle P, T, I, O \rangle$, де P і T — скінчені множини позицій і переходів, I і O — множини вхідних і вихідних функцій.

Іншими словами, мережа Петрі являє собою дводольний орієнтований граф, у якому *позиціям* відповідають вершини, зображені кружками, а *переходам* — вершини, зображені потовщеними рисками; функціям I відповідають дуги, спрямовані від позицій до переходів, а функціям O — від переходів до позицій.

Як і в системах масового обслуговування, у мережах Петрі вводять об'єкти двох типів: *динамічні* — зображуються мітками (маркерами) всередині позицій, і *статичні* — їм відповідають вершини мережі Петрі.

Розподіл маркерів по позиціях називають *маркуванням*. Маркери можуть переміщатися в мережі. Кожна зміна маркування називають *подією*, причому кожна подія зв'язана з певним переходом. Вважається, що події відбуваються миттєво і одночасно при виконанні деяких умов.

Кожній умові в мережі Петрі відповідає визначена позиція. Здійсненню події відповідає *спрацьовування* (або запуск) переходу, при якому маркери з вхідних позицій цього переходу переміщуються у вихідні позиції. Послідовність подій утворить модельований процес.

Правила спрацьовування переходів (рис. 3.21),:

перехід спрацьовує, якщо для кожної з його вхідних позицій виконується умова $N_i \geq K_i$, де N_i — число маркерів у вхідній позиції, K_i — число дуг, що йдуть від i -ї позиції до переходу. При спрацьовуванні переходу число маркерів у i -й вхідній позиції зменшується на K_i , а в j -й вихідній позиції збільшується на M_j , де M_j — число дуг, що з'єднують перехід з j -ю позицією.

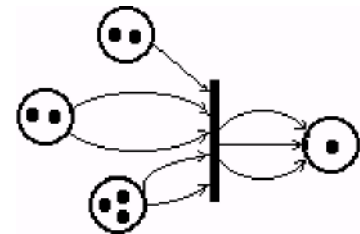


Рис.3.21. Фрагмент мережі Петрі

На рис. 3.21 показано приклад розподілу маркерів по позиціях перед спрацьовуванням, це

систем

маркування записують у виді (2,2,3,1). Після спрацьовування переходу маркування стає: (1,0,1,4).

Можна вводити ряд додаткових правил і умов в алгоритми моделювання і одержувати той або інший різновид мереж Петрі. Так, якщо ввести модельний час, то можна моделювати не тільки послідовність подій, але і їхню часову прив'язку. Це здійснюється наданням переходам ваги — тривалості (затримки) спрацьовування, яку можна визначати, використовуючи певний алгоритм. Отриману модель називають *часовою мережею Петрі*.

Якщо затримки є випадковими величинами, то мережу називають *стохастичною*. У стохастичних мережах можливе введення імовірностей спрацьовування збуджених переходів.

Так, на рис. 3.22 представлений фрагмент мережі Петрі, що ілюструє конфліктну ситуацію — маркер у позиції може запустити або перехід t_1 або перехід t_2 . У стохастичній мережі передбачається можливий вибір переходу, що спрацьовує, у таких ситуаціях.

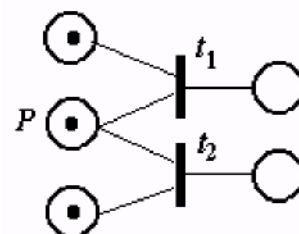


Рис.3.22. Конфліктна ситуація:

Якщо затримки визначаються як функції деяких аргументів, якими можуть бути кількості маркерів у яких-небудь позиціях, стану деяких переходів і т.п., то мережу називають *функціональною*.

У багатьох задачах динамічні об'єкти можуть бути декількох типів, і для кожного типу потрібно вводити свої алгоритми поведінки в мережі. У цьому випадку кожен маркер повинний мати хоча б один параметр, що позначає тип маркера. Такий параметр звичайно називають кольором; колір можна використовувати як аргумент у функціональних мережах. Мережу Петрі при цьому називають *кольоровою*.

Серед інших різновидів мереж Петрі варто згадати *інгібіторні* мережі, що характеризуються тим, що в них можна застосовувати забороняючі (інгібіторні) дуги. Наявність маркера у вхідній позиції, зв'язаної з переходом інгібіторною дугою, означає заборону на спрацьовування переходу.

Приклад 1.

Потрібно описати за допомогою мережі Петрі роботу групи користувачів на єдиній робочій станції WS при заданих характеристиках потоку запитів на користування WS і характеристиках задач, що надходять.

Мережа Петрі представлена на рис. 3.23.

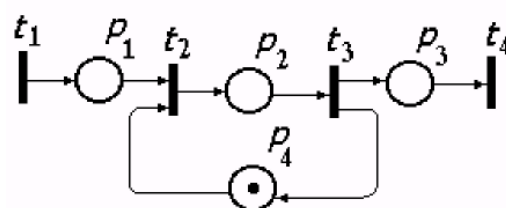


Рис. 3.23. Мережа Петрі для прикладу 1:

Тут переходи зв'язані з наступними подіями: t_1 — надходження запиту на використання WS, t_2 — занятість станції, t_3 — звільнення станції, t_4 — вихід обслуженої заявки; позиція p_4 використовується для відображення стану WS: якщо в p_4 є мітка, то WS вільна і заявка, що прийшла, викликає спрацьовування переходу t_2 ; поки ця заявка не буде обслужена, мітки в p_4 не буде, отже запити, що прийшли в позицію p_1 змушені очікувати спрацьовування переходу t_3 .

Лекція 9. Технічне забезпечення процесу проектування

Приклад 2.

Потрібно описати за допомогою мережі Петрі процеси виникнення й усунення несправностей у деякій технічній системі, що складається з M однотипних блоків; у запасі є один справний блок; відомі статистичні дані про інтенсивності виникнення відмов і тривалості таких операцій, як пошук несправності, заміна і ремонт блоку, що відмовив. На рис. 3.24 представлена відповідна мережа Петрі. Відзначимо, що при кількості міток у позиції, рівній M , можна не ставити M крапок, а записати значення M .

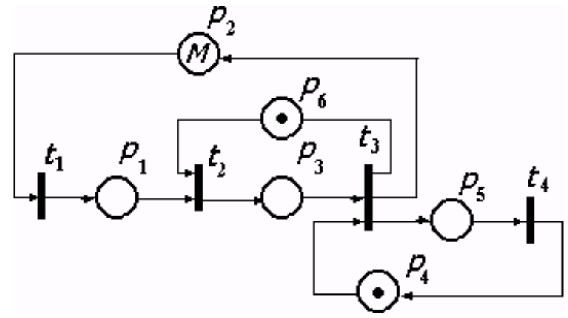


Рис. 3.24. Мережа Петрі для прикладу 2

У нашому прикладі значення M в позиції p_2 відповідає числу наявних у системі блоків. Переходи відображають наступні події: t_1 — відмова блоку, t_2 — пошук несправного блоку, t_3 — його заміна, t_4 — закінчення ремонту.

Очевидно, що при непорожній позиції p_2 перехід t_1 спрацьовує, але з затримкою, рівною обчисленому випадковому значенню модельованого відрізка часу між відмовами. Після виходу маркера з t_1 , він попадає через p_1 в p_2 , якщо є мітка в позиції p_6 . Це означає, що бригада фахівців, що обслуговує систему, вільна і може приступити до пошуку виниклої несправності. У переході t_2 мітка затримується на час, рівний випадковому значенню тривалості пошуку несправності. Далі маркер виявляється в p_3 , якщо є запасний блок (маркер в p_4), то запускається перехід t_3 , з якого маркери вийдуть у p_2 , p_5 і в p_6 через відрізок часу, необхідний для заміни блоку. Після цього в t_4 імітується відновлення несправного блоку.

Розглянута модель описує функціонування системи в умовах, коли відмови можуть виникати і у робочому, і в несправному станах системи. Тому, не виключені ситуації, при яких більш ніж один маркер виявиться в позиції p_1 .

Аналіз мереж Петрі.

Аналіз складних систем на базі мереж Петрі можна виконувати за допомогою імітаційного моделювання СМО, представлених моделями мереж Петрі. При цьому задають вхідні потоки заявок і визначають відповідну реакцію системи. Вихідні параметри СМО розраховують шляхом обробки накопиченого при моделюванні статистичного матеріалу. Можливий і інший підхід до використання мереж Петрі для аналізу об'єктів, досліджуваних на системному рівні. Він не зв'язаний з імітацією процесів і заснований на дослідженні таких властивостей мереж Петрі, як *обмеженість*, *безпечність*, *збережуваність*, *досяжність*.

Обмеженість має місце, якщо кількість міток у будь-якій позиції мережі не може перевищити значення K . При проектуванні автоматизованих систем визначення K дозволяє обґрунтовано вибирати ємності нагромаджувачів. Можливість необмеженого росту числа міток свідчить про небезпеку необмеженого росту довжин черг.

Безпечність — окремий випадок обмеженості, а саме це 1-обмеженість. Якщо для деякої позиції встановлено, що вона безпечна, то її можна представляти одним тригером.

Збережуваність характеризується сталістю завантаження ресурсів, тобто

$$\sum A_i N_i = \text{const},$$

де N_i — число маркерів у i -й позиції, A_i — ваговий коефіцієнт.

Досяжність $M_k \rightarrow M_j$ характеризується можливістю досягнення маркування M_j зі стану мережі, що характеризується маркуванням M_k .

В основі дослідження перерахованих властивостей мереж Петрі лежить аналіз досяжності.

Один з методів аналізу досяжності будь-якого маркування зі стану M_0 — побудова графа досяжності. Початкова вершина графа відображає M_0 , а інші вершини відповідають маркуванням. Дуга з M_i у M_j означає подію $M_i \rightarrow M_j$ і відповідає спрацьовуванню переходу t . У складних мережах граф може містити надмірно велике число вершин і дуг. Однак при побудові графа можна не відображати усі

систем

вершини, тому що багато з них є дублями (дійсно, від маркування M_k завжди породжується той самий підграф не залежно від того, з якого стану система прийшла в M_k). Тупіки виявляються по відсутності дозволених переходів з якої-небудь вершини, тобто по наявності листівів — термінальних вершин. Необмежений ріст числа маркерів у якій-небудь позиції свідчить про порушення обмеженості.

Приклади аналізу досяжності.

Приклад 1. Мережа Петрі і граф досяжних розміток представлені на рис. 3.25.

На малюнку вершини графа зображені у виді маркірувань, дуги позначені переходами, що спрацьовують. Жвавість мережі очевидна, тому що спрацьовують усі переходи, тупіки відсутні.

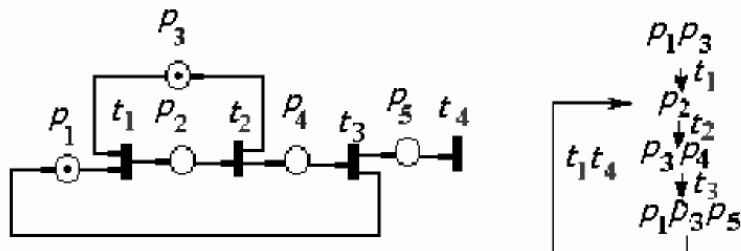


Рис. 3.25. Мережа Петрі і її граф досяжності для прикладу 1

Приклад 2. Мережа Петрі і граф досяжних розміток представлені на рис. 3.26.

Мережа, що моделює двупроцесорну обчислювальну систему з загальною пам'яттю, є живою, усі розмітки досяжні.

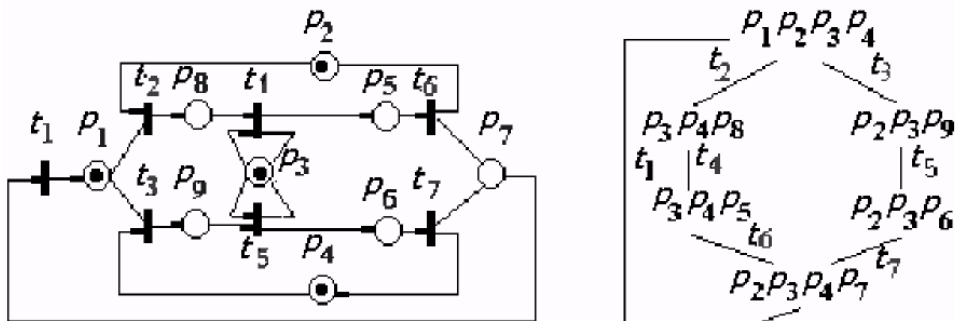


Рис. 3.26. Мережа Петрі і її граф досяжності для прикладу 2

19. Математичне забезпечення підсистем машинної графіки і геометричного моделювання

Компоненти математичного забезпечення.

Підсистеми машинної графіки і геометричного моделювання (МГІГМ) займають центральне місце в машинобудівних САПР-К. Конструювання виробів у них, як правило, проводиться в інтерактивному режимі при оперуванні геометричними моделями, тобто математичними об'єктами, що відображають форму деталей, склад збірних вузлів і, можливо, деякі додаткові параметри (маса, момент інерції, кольори поверхні і т.п.).

У підсистемах МГІГМ типовий маршрут обробки даних передбачає одержання проектного рішення в прикладній програмі і його представлення у виді геометричної моделі (геометричне моделювання), далі підготовку проектного рішення до візуалізації, власне візуалізацію в апаратурі робочої станції і, при необхідності, коригування рішення в інтерактивному режимі. Дві останні операції реалізуються на базі апаратних засобів машинної графіки. Коли говорять про *математичне забезпечення МГІГМ*, мають на увазі, насамперед, моделі, методи й алгоритми для геометричного моделювання і підготовки до візуалізації. При цьому часто саме математичне забезпечення підготовки до візуалізації називають математичним забезпеченням машинної графіки.

Розрізняють математичне забезпечення двовимірною (2) і тривимірною (3) моделювання. Основні застосування 2D графіки — підготовка креслярської документації в машинобудівних САПР, топологічне проектування друкованих плат і кристалів БІС у САПР електронної промисловості. У розвинутих машинобудівних САПР використовують як 2D, так і 3D моделювання для синтезу конструкцій, представлення траєкторій робочих органів верстатів при обробці заготовель, генерації сітки скінчених елементів при аналізі міцності і т.п.

У 3D моделюванні розрізняють моделі **каркасні** (дротові), **поверхневі**, **об'ємні** (твердотілі).

Каркасна модель представляє форму деталі у виді скінченої множини ліній, що лежать на поверхнях деталі. Для кожної лінії відомі координати скінчених точок і зазначена їх інцидентність ребрам або поверхням. Оперувати каркасною моделлю на подальших операціях маршрутів проектування незручно, і тому каркасні моделі в даний час використовують рідко.

Поверхнева модель відображає форму деталі за допомогою задання обмежуючих її поверхонь, наприклад, у виді сукупності даних про грані, ребра і вершини.

Особливе місце займають моделі деталей з поверхнями складної форми, так звані скульптурними поверхнями. До таких деталей відносяться корпуси багатьох транспортних засобів (напр., суден, автомобілів), деталі, обтічні потоками рідин і газів (лопатки турбін, крила літаків), й ін.

Об'ємні моделі відрізняються тим, що в них у явній формі утримуються дані про приналежність елементів внутрішньої або зовнішнього стосовно деталі просторові. В даний час застосовують наступні підходи до побудови геометричних моделей.

1. Завдання граничних елементів — граней, ребер, вершин.
2. *Кінематичний метод*, відповідно до якого задають двовимірний контур і траєкторію його переміщення; слід від переміщення контуру приймають як поверхню деталі.
3. *Позиційний підхід*, відповідно до якого розглянутий простір розбивають на комірки (позиції) і деталь задають за допомогою переліку комірок, що належать деталі — очевидна громіздкість цього підходу.
4. Представлення складної деталі у виді сукупностей *базових елементів форми* (БЕФ) і виконуваних над ними теоретико-множинних операцій. До БЕФ відносяться заздалегідь розроблені моделі простих тіл, це, у першу чергу, моделі паралелепіпеда, циліндра, сфери, призми. Типовими теоретико-множинними операціями є об'єднання, перетинання, різниця. Наприклад, модель плити з отвором у ній може бути отримана шляхом вилучення циліндра з паралелепіпеда.

Метод на основі БЕФ часто називають *методом конструктивної геометрії*. Це основний спосіб

конструювання складальних вузлів у сучасних САПР-К.

У пам'яті ЕОМ всі геометричні моделі зазвичай зберігаються у векторній формі, тобто у виді координат сукупності точок, що задають елементи моделі. Виконання операцій конструювання також виконується над моделями у векторній формі. Найбільш компактна модель – у виді сукупності зв'язаних БЕФ, що переважно і використовується для збереження й обробки інформації про вироби в системах конструктивної геометрії.

Однак для візуалізації в сучасних робочих станціях у зв'язку з використанням у них растрових дисплеїв *необхідна растеризація* — перетворення моделі в растрову форму. Зворотню операцію переходу до векторної форми, що характеризується меншими витратами пам'яті, називають *векторизацією*. Зокрема, векторизація повинна виконуватися стосовно даних, одержуваним скануванням зображень.

<навести приклади та характеристики модулів, які здійснюють такі операції>

Геометричні моделі.

Важливою складовою частиною геометричних моделей є опис поверхонь. Якщо поверхні деталі — плоскі грані, то модель може бути задана досить просто, а саме – інформацією про грані, ребра та вершини деталі. При цьому звичайно використовується метод конструктивної геометрії. Представлення за допомогою плоских граней має місце й у випадку більш складних поверхонь, якщо ці поверхні апроксимувати множинами плоских ділянок — полігональними сітками. Тоді можна поверхневу модель задати однією з наступних форм:

1) модель у вигляді списку граней, де кожна грань представлена упорядкованим списком вершин (циклом вершин); ця форма характеризується значною надмірністю, тому що кожна вершина повторюється в декількох списках.

2) модель у вигляді списку ребер, де для кожного ребра задані інцидентні вершини і грані.

Однак апроксимація полігональними сітками при великих розмірах елементів сітки дає помітні перекручування форми, а при малих розмірах осередків виявляється неефективною по обчислювальних витратах. Тому більш популярні описи неплоских поверхонь кубічними рівняннями у формі Без'є або b-сплайнів.

Знайомство з цими формами зручно виконати, показавши їхнє застосування для опису геометричних об'єктів першого рівня — просторових кривих.

Геометричними об'єктами нульового, першого і другого рівнів називають відповідно точки, криві, поверхні.

У підсистемах МГіГМ використовуються кубічні криві, що задаються параметрично:

$$\begin{aligned}x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x, \\y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y, \\z(t) &= a_z t^3 + b_z t^2 + c_z t + d_z,\end{aligned} \quad (3.48)$$

де $1 \leq t \leq 0$. Такими кривими описують сегменти апроксимованій кривої, тобто апроксимовану криву розбивають на сегменти і кожен сегмент апроксимують рівняннями (3.48).

Застосування кубічних кривих забезпечує (відповідним вибором чотирьох коефіцієнтів у кожному із трьох рівнянь) виконання чотирьох умов сполучення сегментів. У випадку кривих Без'є цими умовами є проходження кривої сегмента через дві задані кінцеві точки і рівність у цих точках дотичних

212Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Лекція 9. Технічне забезпечення процесу проектування

векторів сусідніх сегментів. У випадку b-сплайнів виконуються умови безперервності дотичного вектора і кривизни (тобто першої і другої похідних) у двох скінчених точках, що забезпечує високий ступінь "гладкості" кривій, хоча проходження апроксимуючої кривої через задані крапки тут не забезпечується. Застосування поліномів вище третього ступеня не рекомендується, тому що велика імовірність появи неоправданої "хвилястості".

У випадку форми Без'є коефіцієнти в (3.48) визначаються, по-перше, підстановкою в (3.48) значень $t=0$ і $t=1$ і координат заданих скінчених точок P_1 і P_4 відповідно, по-друге, підстановкою у

$$\begin{aligned} dx/dt &= 3a_x t^2 + 2b_x + c_x, \\ dy/dt &= 3a_y t^2 + 2b_y + c_y, \\ dz/dt &= 3a_z t^2 + 2b_z + c_z, \end{aligned}$$

виразах похідних

тих же значень $t=0$ і $t=1$ і координат точок P_2 і P_3 , що задає напрямки дотичних векторів (рис. 3.27). У

$$\begin{aligned} x(t) &= \mathbf{T}^T \mathbf{M} \mathbf{G}_x, \\ y(t) &= \mathbf{T}^T \mathbf{M} \mathbf{G}_y, \\ z(t) &= \mathbf{T}^T \mathbf{M} \mathbf{G}_z, \end{aligned} \quad (3.49)$$

де $\mathbf{T}^T = (t^3, t^2, t, 1)$ — вектор-рядок, матриця \mathbf{M} представлена в табл. 3.11, \mathbf{G}_x — вектор координат P_{xi} точок P_1, P_2, P_3 і P_4 , аналогічно $\mathbf{G}_y, \mathbf{G}_z$ вектори координат P_{yi}, P_{zi} тих же точок.

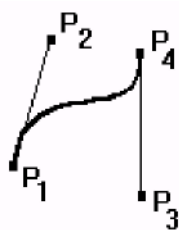


Рис. 3.27. Крива Без'є

-1	3	-3	1
3	-6	3	0
-3	3	0	0
1	0	0	0

Таблиця 3.11

-1/6	1/2	-1/2	1/6
1/2	-1	1/2	0
-1/2	0	1/2	0
1/6	2/3	1/6	0

Таблиця 3.12

У випадку b-сплайнів апроксимована крива поділяється на n ділянок, які виділяються послідовними точками $P_0, P_1, P_2, \dots, P_n$. Ділянка між парою сусідніх точок P_i і P_{i+1} апроксимується b-сплайном, побудованим з використанням чотирьох точок $P_{i-1}, P_i, P_{i+1}, P_{i+2}$. b-сплайн на ділянці $[P_i, P_{i+1}]$ може бути представлений виразом, аналогічними (3.49):

$$\begin{aligned} x(t) &= \mathbf{T}^T \mathbf{M} \mathbf{G}_x, \\ y(t) &= \mathbf{T}^T \mathbf{M} \mathbf{G}_y, \\ z(t) &= \mathbf{T}^T \mathbf{M} \mathbf{G}_z, \end{aligned}$$

для якого матриця \mathbf{M} має інший вид і представлений у табл. 3.12, а вектори $\mathbf{G}_x, \mathbf{G}_y, \mathbf{G}_z$ містять відповідні координати точок $P_{i-1}, P_i, P_{i+1}, P_{i+2}$.

Аналогічно можна отримати вирази для форм Без'є та b-сплайнів стосовно поверхонь, але з урахуванням того, що замість (3.48) використовують кубічні залежності від двох змінних.

Методи й алгоритми машинної графіки (підготовки до візуалізації).

До методів машинної графіки відносять також методи перетворення графічних об'єктів, представлення (розгорнення) ліній у растровій формі, виділення вікна, видалення схованих ліній, проектування, зафарбовування зображень.

Перетворення графічних об'єктів виконуються за допомогою операцій переносу, масштабування, повороту. Перенос точки з положення P у нове положення S можна виконувати по формулах типу

$$C_{xi} = P_{xi} + \Delta x_i,$$

де Δx_i — приріст по координаті x_i . Однак зручніше операції перетворення представляти в єдиній матричній формі

$$C = P T, \quad (3.50)$$

де T — перетворююча матриця. При цьому точки C і P у двовимірному випадку зображують векторами-рядками 1×3 , у яких крім значень двох координат (які називають при такому представленні однорідними), додатково вказують масштабний множник W . Тоді перенос для випадку $2D$ можна виразити у виді (3.50), де T є табл. 3.13, а $W=1$.

Для операцій масштабування і повороту, матриці T представлені в табл. 3.14 і 3.15, відповідно. Тут m_x , m_y — масштабні множники, φ — кут повороту.

Таблиця 3.13			Таблиця 3.14			Таблиця 3.15		
1	0	0	m_x	0	0	cos φ	sin φ	0
0	1	0	0	m_y	0	- sin φ	cos φ	0
Δx_1	Δx_2	1	0	0	1	0	0	1

Зручність (3.50) обумовлена тим, що будь-яку комбінацію елементарних перетворень можна описати формулою (3.50). Наприклад, вираз для переміщення з одночасним поворотом має вигляд

$$C = P T_{\text{сд}} T_{\text{пов}} = P T,$$

де $T = T_{\text{сд}} T_{\text{пов}}$, $T_{\text{сд}}$ — матриця зрушення, $T_{\text{пов}}$ — матриця повороту.

Представлення графічних елементів у растровій формі потрібно для відображення цих елементів на бітову карту растрової відеосистеми. Нехай потрібно розгорнути відрізок AB прямої $y = ax + b$, причому $(0 \leq a \leq 1)$. Введемо позначення: $A = (x_a, y_a)$, $B = (x_b, y_b)$; за величину дискрета (пікселя) прийемо одиницю. В алгоритмі розгорнення номера рядків і стовпців карти, на перетинанні яких повинні знаходитися точки відрізка, визначаються в такий спосіб:

- 1) $\Delta x := x_b - x_a$; $\Delta y := y_b - y_a$; $x := x_a$; $y := y_a$;
- 2) $d := 2\Delta y - \Delta x$;
- 3) якщо $d \geq 0$, тоді $\{y := y + 1, d := d + 2(\Delta y - \Delta x)\}$; інакше $d := d + 2\Delta y$;
- 4) $x := x + 1$;
- 5) перехід до пункту 3, поки не досягнута точка B .

Цей же алгоритм із невеликими модифікаціями використовується і при інших значеннях коефіцієнта a .

Економічність цього алгоритму обумовлюється відсутністю довгих арифметичних операцій типу множення.

Виділення вікна потрібно при визначенні тієї частини сцени, що повинна бути виведена на екран дисплея.

Лекція 9. Технічне забезпечення процесу проектування

Нехай вікно обмежене лініями $x = x_1$, $x = x_2$, $y = y_1$, $y = y_2$ (рис. 3.28). По черзі для кожного багатокутника перевіряється розташування його вершин і ребер щодо границь вікна.

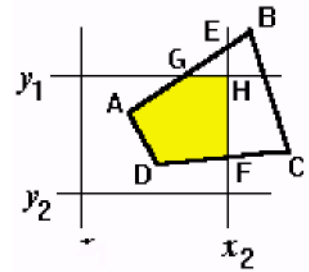


Рис. Виділення вікна.

Так, для багатокутника **ABCD** (див. рис. 3.28) при відсіканні по границі $x = x_2$ проглядається безліч вершин у порядку обходу по годинній стрілці.

Можливі чотири ситуації для двох послідовних вершин **P** і **R**:

- 1) якщо $x_p > x_2$ і $y_p > y_2$, обидві вершини і інцидентне їм ребро знаходяться поза вікном і виключаються з подальшого аналізу;
- 2) якщо $x_p < x_2$ і $x_r < x_2$, то обидві вершини і інцидентне їм ребро залишаються для подальшого аналізу;
- 3) якщо $x_p < x_2$ і $x_r > x_2$, то вершина **P** залишається в списку вершин, а вершина **R** замінюється новою вершиною з координатами $x = x_2$, $y = y_p + (y_r - y_p)(x_2 - x_p)/(x_r - x_p)$, у нашому прикладі такої нової вершиною буде **E**;
- 4) якщо $x_p > x_2$ і $y_p < y_2$, то вершина **P** замінюється новою вершиною з координатами $x = x_2$, $y = y_p - (y_r - y_p)(x_2 - x_p)/(x_r - x_p)$, а вершина **R** залишається в списку вершин; у нашому прикладі нової вершини буде **F**.

Після закінчення перегляду стосовно до всіх границь у вікні виявляються, що залишилися в списку вершини.

Застосовуючи ці правила в нашому прикладі, одержуємо спочатку багатокутник **AEFD**, а після відсікання по верхній границі — багатокутник **AGFD** (див. рис. 3.28). Однак правильний результат трохи інший, а саме багатокутник **AGHFD**. Цей правильний результат виходить при подвійному обході вершин спочатку по годинній стрілці, потім проти з включенням у список нових вершин, що з'являються при кожному обході.

Застосовують ряд алгоритмів *видалення схованих ліній*. Один з найбільше просто реалізованих алгоритмів — алгоритм z-буфера, де z-буфер — область пам'яті, число осередків у якій дорівнює числу пікселів у вікні висновку. Передбачається, що вісь **Z** спрямована по нормалі до видової поверхні і спостерігач розташований у точці $Z = 0$.

На початку виконання алгоритму всі пікселі відповідають максимальному значенню **Z**, тобто максимальному видаленню від спостерігача, що приводить до приміщення в усі осередки z-буфера значень пікселів тла картини (креслення). Далі по черзі для всіх точок граней розраховуються значення координати **Z**. Серед точок, що відносяться тому самому пікселю (однієї і тому ж осередковій z-буфера **S**), вибирається точка з найменшим значенням **Z** і її код (тобто колір і яскравість) міститься в **S**. У підсумку z-буфер буде містити пікселі найбільш близьких до спостерігача граней.

Алгоритми побудови проєкцій перетворюють тривимірні зображення в двовимірні. У випадку побудови центральної проєкції кожна точка тривимірного зображення відображається на картинну поверхню шляхом перерахування координат x і y (рис. 3.29).

Так, координату x_a' точки **A** обчислюють по очевидній формулі:

$$x_a' = x_a d/z,$$

аналогічно розраховується координата y_a точки **A'**.

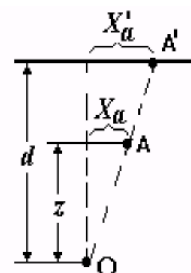


Рис. 3.29. Побудова центральної проєкції

У рівнобіжних проекціях $d \rightarrow \infty$ і координати x і y у точках A' і A збігаються. Тому побудова рівнобіжних проекцій зводиться до виділення вікна, при необхідності до повороту зображення і можливо до видалення схованих ліній.

Зафарбування матових поверхонь засновані на законі Ламберта, відповідно до якого яскравість відбитого від поверхні світла пропорційні $\cos \alpha$, де α — кут між нормаллю до поверхні і напрямком променя падаючого світла. В алгоритмі Гуро яскравість внутрішніх точок визначається лінійною інтерполяцією яскравості у вершинах багатокутника. При цьому спочатку проводиться інтерполяція в точках ребер, а потім по рядках горизонтального розгорнення. Більш реалістичними виходять зображення в алгоритмі Фонга, заснованому на лінійній інтерполяції векторів нормалей до поверхні.

ПК геометричного моделювання і машинної графіки.

Світовими лідерами в цій області програмного забезпечення САПР вважаються Pro/Engineer, Unigraphics, EUCLID, CADD5, CATIA, I-DEAS і ряд інших. По своїх функціональних можливостях ці комплекси приблизно рівноцінні: нові елементи, що з'явилися в одному з них, незабаром реалізуються в нових версіях інших комплексів. Тому, для першого знайомства, достатньо розглянути характеристики одного з комплексів. Нижче приведені коротка інформація по Pro/Engineer.

Комплекс нараховує кілька десятків програм (модулів), що розділені на групи програм конструкторського проектування механічних об'єктів, промислового дизайну, функціонального моделювання, технологічного проектування, обміну даними.

Базові модулі конструкторського проектування призначені для твердотілого і поверхневого моделювання, синтезу конструкцій з базових елементів форми, підтримують параметризацію й асоціативність, проекційне креслення, виконують розробку креслень із простановкою розмірів і допусків. Користувач може поповнювати бібліотеку БЭФ оригінальними моделями. Синтез тривимірних моделей складної форми можливий витягуванням плоского контуру по нормалі до його площини, його протяганням уздовж довільній просторовій кривій, обертанням контуру навколо заданої осі, натягуванням між декількома заданими перетинами. Синтез зборок виконується викликом або посиланням на бібліотечні елементи, їхньою модифікацією, розробкою нових деталей. Деталі зборки можна потрібним чином орієнтувати в просторі. Далі варто ввести асоціативні зв'язки.

Додаткові модулі конструкторського проектування мають більш конкретну, але вузьку спеціалізацію. Прикладами таких модулів можуть служити модулі конструювання панелей з композитних матеріалів, розробки штампів і ливарних прес-форм, трубопровідних систем, зварених конструкцій, розведення електричних кабелів і дротів.

Модулі функціонального моделювання використовуються як препроцесори і постпроцесори для програм кінцево-елементного аналізу (нанесення сітки скінчених елементів, візуалізація результатів аналізу), для аналізу теплового стану конструкцій, для оцінки вібровитривалості й ін.

Основні модулі технологічного проектування служать для моделювання технологічних процесів фрезерної, токарської, електроерозійної обробки і для розробки постпроцесорів для систем керування устаткуванням із ЧПУ. Серед модулів обміну даними важлива наявність взаємозв'язків по стандарті STEP, що відкриває можливості імпорту/експорту даних з різними CAE/CAD/CAM системами, що підтримують цей стандарт.

20. Математичне забезпечення параметричного синтезу в АП СОС

Постановка задач параметричного синтезу

Місце процедур синтезу в проектуванні

Сутність проектування полягає в прийнятті проектних рішень, що забезпечують виконання майбутнім об'єктом пропонованих до нього вимог. Синтез проектних рішень — основа проектування; від успішного виконання процедури синтезу у визначальній мері залежать споживчі властивості майбутньої продукції. Звичайно, аналіз — необхідна складова частина проектування, що служить для верифікації прийнятих проектних рішень. Саме аналіз дозволяє одержати необхідну інформацію для цілеспрямованого виконання процедур синтезу в ітераційному процесі проектування. Тому синтез і аналіз нерозривно зв'язані.

Як відзначено в гл. 1, синтез підрозділяють на параметричний і структурний. Проектування починається зі *структурного синтезу*, при якому генерується принципове рішення. Таким рішенням може бути вигляд майбутнього літального апарата, або фізичний принцип дії датчика, або одна з типових конструкцій двигуна, або функціональна схема мікропроцесора. Але ці конструкції і схеми вибирають у параметричному виді, тобто без указівки числових значень параметрів елементів. Тому перш ніж приступити до верифікації проектного рішення, потрібно задати або розрахувати значення цих параметрів, тобто виконати *параметричний синтез*. Прикладами результатів параметричного синтезу можуть служити геометричні розміри деталей у механічному вузлі або в оптичному приладі, параметри електро-радіо-елементів в електронній схемі, параметри режимів різання в технологічній операції і т.п.

У випадку якщо за результатами аналізу проектного рішення визнається неостаточним, то починається процес послідовних наближень до прийнятного варіанта проекту. У багатьох додатках для поліпшення проекту зручніше варіювати значення параметрів елементів, тобто використовувати параметричний синтез на базі різноманітного аналізу. При цьому задача параметричного синтезу може бути сформульована як задача визначення значень параметрів елементів, найкращих з позицій задоволення вимог технічного завдання при незмінній структурі проектного об'єкта. Тоді параметричний синтез називають параметричною оптимізацією або просто *оптимізацією*. Якщо параметричний синтез не приводить до успіху, то повторюють процедури структурного синтезу, тобто на чергових ітераціях коректують або переобирають структуру об'єкта.

Критерії оптимальності.

У САПР процедури параметричного синтезу виконуються або людиною в процесі різноманітного аналізу (в інтерактивному режимі), або реалізуються на базі формальних методів оптимізації (в автоматичному режимі). В останньому випадку знаходять застосування кілька постановок задач оптимізації.

Найбільш розповсюдженою є детермінована постановка: задані умови працездатності на вихідні параметри Y і потрібно знайти номінальні значення проектних параметрів X , до яких відносяться параметри усіх або частини елементів проектного об'єкта. Назвемо цю задачу оптимізації *базовою*. В окремому випадку, коли вимоги до вихідних параметрів задані нечітко, до числа величин, що розраховуються, можуть бути віднесені також норми вихідних параметрів, що фігурують у їхніх умовах працездатності.

Якщо проектуються вироби для подальшого серійного виробництва, то важливого значення набуває такий показник, як відсоток випуску придатних виробів у процесі виробництва. Очевидно, що успішне виконання умов працездатності в номінальному режимі не гарантує їхнього виконання при обліку

систем

виробничих погрішностей, що задаються допусками параметрів елементів. Тому метою оптимізації стає максимізація відсотка виходу придатних, а до результатів рішення задачі оптимізації відносяться не тільки номінальні значення проектних параметрів, але і їхні допуски.

Базова задача оптимізації ставиться як задача математичного програмування:

$$\begin{aligned} & \text{extr}_{\mathbf{X} \in \mathbf{D}_x} F(\mathbf{X}), \\ & \mathbf{D}_x = \{\mathbf{X} \mid \varphi(\mathbf{X}) > 0, \psi(\mathbf{X}) = 0\}, \end{aligned} \quad (4.1)$$

де $F(\mathbf{X})$ — цільова функція, \mathbf{X} — вектор керованих (проектних) параметрів, $\varphi(\mathbf{X})$ і $\psi(\mathbf{X})$ — функції-обмеження, \mathbf{D}_x -допустимість область у просторі керованих параметрів. Запис (4.1) інтерпретується як задача пошуку екстремума цільової функції шляхом варіювання керованих параметрів у межах припустимої області.

Таким чином, для виконання розрахунку номінальних значень параметрів необхідно, по-перше, сформулювати задачу у виді (4.1), по-друге, вирішити задачу пошуку екстремуму $F(\mathbf{X})$.

Складність постановки оптимізаційних проектних задач обумовлена наявністю в проєктованих об'єктах декількох вихідних параметрів, що можуть бути критеріями оптимальності, але в задачі (4.1) цільова функція повинна бути одна. Іншими словами, проектні задачі є багатокритеріальних, і виникає проблема зведення багатокритеріальної задачі до однокритеріальної.

Застосовують кілька способів вибору критерію оптимальності.

У частковому критерії серед вихідних параметрів один вибирають як цільову функцію, а умови працездатності інших вихідних параметрів відносять до обмежень задачі (4.1). Ця постановка цілком прийнятна, якщо дійсно можна виділити один найбільш критичний вихідний параметр. Але в більшості випадків позначається недолік приватного критерію (мал. 4.1).

На цьому малюнку представлений двовимірний простір вихідних параметрів y_1 і y_2 , для яких задані умови працездатності $y_1 < T_1$ і $y_2 < T_2$. Крива АВ є границею досяжних значень вихідних параметрів. Це обмеження об'єктивне і зв'язане з існуючими фізичними і технологічними умовами виробництва, називаними умовами реалізації. Область, у межах якої виконуються всі умови реалізації і працездатності, називають областю працездатності.



Рис. 4.1. Області Парето і працездатності

Безліч точок простору вихідних параметрів, з яких неможливе переміщення, що приводить до поліпшення усіх вихідних параметрів, називають областю компромісів, або областю Парето. Ділянка кривій АВ (див. мал. 4.1) відноситься до області Парето.

Якщо як цільову функцію в ситуації мал. 4.1. вибрати параметр y_1 те результатом оптимізації будуть параметри \mathbf{X} , що відповідають точці В. Але це границя області працездатності і, отже, при нестабільності внутрішніх і зовнішніх параметрів велика імовірність виходу за межі області працездатності. Звичайно, результати можна поліпшити, якщо застосовувати так називаний метод поступок, при якому як обмеження приймають умова працездатності зі скоректованою нормою у виді

$$y_2 < T_2 + \Delta,$$

де А — поступка. Але виникає проблема вибору значень поступок, тобто результати оптимізації будуть мати суб'єктивний характер. Очевидно, що ситуація не зміниться, якщо цільовою функцією

218Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Лекція 9. Технічне забезпечення процесу проектування

буде обраний параметр y_2 , — оптимізація приведе в крапку А.

Адитивний критерій поєднує (згортає) усі вихідні параметри (частки критерії) в одну цільову функцію, що представляє собою зважену суму приватних критеріїв

$$F(\mathbf{X}) = \sum_{j=1}^m \omega_j y_j(\mathbf{X}), \quad (4.2)$$

де ω_j — вагарні коефіцієнт, m — число вихідних параметрів. Функція (4.2) підлягає мінімізації, при цьому якщо умова працездатності має вигляд $y_j > T_j$, то $\omega_j < 0$

Недоліки адитивного критерію — суб'єктивний підхід до вибору вагових коефіцієнтів і неврахування вимог ТЗ. Дійсно в (4.2) не входять норми вихідних параметрів.

Аналогічні недоліки властиві і *мультиплікативному критерію*, цільова функція якого має вигляд

$$F(\mathbf{X}) = \prod_{j=1}^m y_j^{\omega_j}(\mathbf{X}). \quad (4.3)$$

Неважно побачити, що якщо прологарифмувати (4.3), то мультиплікативний критерій перетворюється в адитивний.

Більш кращим є *максимальний критерій*, як цільову функцію якого приймають вихідний параметр, найбільш неблагополучний з позицій виконання умов працездатності. Для оцінки ступеня виконання умови працездатності j -го вихідного параметра вводять запас працездатності цього параметра S_j і цей запас можна розглядати як нормований j -й вихідний параметр. Наприклад (тут і далі для лаконічності викладу передбачається, що усі вихідні параметри приведені до виду, при якому умови працездатності стають нерівностями у формі $y_j < T_j$):

$$S_j = (T_j - y_j) / T_j$$

або

$$S_j = (T_j - y_{\text{ном}j}) / \delta_j,$$

де $y_{\text{ном}j}$ -номінальне значення, а δ_j -деяка характеристика розсіювання j -го вихідного параметра, наприклад, трохсмуговий допуск. Тоді цільова функція в максимальному критерії є

$$F(\mathbf{X}) = \min_{j \in [1:m]} Z_j(\mathbf{X}).$$

Тут запис $[1: m]$ означає безліч цілих чисел у діапазоні від 1 до m . Задача (4.1) при максимальним критерії конкретизується в такий спосіб:

$$F(\mathbf{X}) = \max_{\mathbf{X} \in \mathbf{D}_x} \min_{j \in [1:m]} Z_j(\mathbf{X}), \quad (4.4) \text{ де припустима область } \mathbf{D}_x \text{ визначається тільки прямими}$$

обмеженнями на керовані параметри x_i :

$$x_{i \min} < x_i < x_{i \max}$$

Задачі оптимізації з урахуванням допусків.

Змістовну сторону оптимізації з урахуванням допусків пояснює мал. 4.2, на якому представлені області працездатності і пропускна у двовимірному просторі керованих параметрів. Якщо власне допуски задані і не відносяться до керованих параметрів, то ціль оптимізації — максимальним образом сполучити ці області так, щоб імовірність виходу за межі області працездатності була мінімальною.

систем



Рис. 4.2. Области допустима і працездатності

Рішення цієї задачі винятково трудомістко, тому що на кожному кроці оптимізації потрібно виконувати оцінку згаданої імовірності методами статистичного аналізу, а для складних моделей об'єктів таким методом є метод статистичних іспитів. Тому на практиці подібні задачі вирішують, приймаючи ті або інші допущення.

Наприклад, якщо допустити, що ціль оптимізації досягається при сполученні центрів областей працездатності \mathcal{E} і пропускну $X_{\text{ном}}$, то оптимізація зводиться до *задачі центрування*, тобто до визначення центра \mathcal{E} . Задачу центрування звичайно вирішують шляхом попереднього нормування керованих параметрів x_i з наступним уписуванням гіперкуба з максимально можливими розмірами в нормовану область працездатності.

Примітка. Нормування проводять таким чином, що пропусчна область здобуває форму гіперкуба, що виходить після нормування.

Очевидно, що рішення задачі центрування дозволяє не тільки оптимізувати номінальні значення проектних параметрів, але і їхні допуски, якщо останні відносяться до керованих параметрів.

21. Огляд методів оптимізації

Класифікація методів математичного програмування.

У САПР основними методами оптимізації є пошукові методи. Пошукові методи засновані на покроковій зміні керованих параметрів

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \Delta \mathbf{X}_k, \quad (4.5)$$

де в більшості методів збільшення $\Delta \mathbf{X}_k$ вектора керованих параметрів обчислюється по формулі

$$\Delta \mathbf{X}_k = h \mathbf{g}(\mathbf{X}_k). \quad (4.6)$$

Тут \mathbf{X}_k — значення вектора керованих параметрів на k -м кроці, h — крок, а $\mathbf{g}(\mathbf{X}_k)$ — напрямок пошуку. Отже, якщо виконуються умови збіжності, то реалізується покрокове (ітераційне) наближення до екстремуму.

Методи оптимізації класифікують по ряду ознак.

У залежності від числа керованих параметрів розрізняють методи *одномірної* і *багатомірної* оптимізації, у перших з них керований параметр єдиний, у других розмір вектора \mathbf{X} не менш двох. Реальні задачі в САПР багатомірні, методи одномірної оптимізації відіграють допоміжну роль на окремих етапах багатомірного пошуку.

Лекція 9. Технічне забезпечення процесу проектування

Розрізняють методи *умовної* і *безумовної* оптимізації по наявності або відсутності обмежень. Для реальних задач характерна наявність обмежень, однак методи безумовної оптимізації також становлять інтерес, оскільки задачі умовної оптимізації за допомогою спеціальних методів можуть бути зведені до задач без обмежень.

У залежності від числа екстремумів розрізняють задачі одно- і багатоекстремальне. Якщо метод орієнтований на визначення якого-небудь локального екстремуму, то такий метод відноситься до *локальних методів*. Якщо ж результатом є глобальний екстремум, то метод називають *методом глобального пошуку*. Задовільні по обчислювальній ефективності методи глобального пошуку для загального випадку відсутні і тому на практиці в САПР використовують методи пошуку локальних екстремумів.

Нарешті, у залежності від того, використовуються при пошуку похідні цільової функції по керованих чи параметрах ні, розрізняють методи декількох порядків. Якщо похідні не використовуються, то має місце метод *нульового порядку*, якщо використовуються перші або другі похідні, те відповідно метод *першого* або *другого порядку*. Методи першого порядку називають також градієнтними, оскільки вектор перших похідних $F(X)$ по X є градієнт цільової функції

$$\text{grad } (F(\mathbf{X})) = (\partial F/\partial x_1, \partial F/\partial x_2, \dots, \partial F/\partial x_n).$$

Конкретні методи визначаються наступними факторами:

1. способом обчислення напрямку пошуку $g(X_k)$ у формулі (4.6);
2. способом вибору кроку h ;
3. способом визначення закінчення пошуку.

Визначальним фактором є перший з перерахованих у цьому списку, він докладно описаний нижче.

Крок може бути або постійним, або вибиратися виходячи з одновимірної оптимізації — пошуку мінімуму цільової функції в обраному напрямку $\mathbf{g}(\mathbf{X}_k)$. В останньому випадку крок будемо називати оптимальним.

Закінчення пошуку звичайно здійснюють за правилом: якщо протягом r підряд йдуть кроків траєкторія пошуку залишається в малій ϵ -околиці поточної крапки пошуку X_k те пошук варто припинити, отже, умова закінчення пошуку має вигляд

$$|\mathbf{X}_k - \mathbf{X}_{k-r}| < \epsilon.$$

Методи одновимірної оптимізації.

До методів одновимірної оптимізації відносяться методи дихотомічного розподілу, золотого перетину, чисел Фібоначі, поліноміальної апроксимації і ряд їхніх модифікацій.

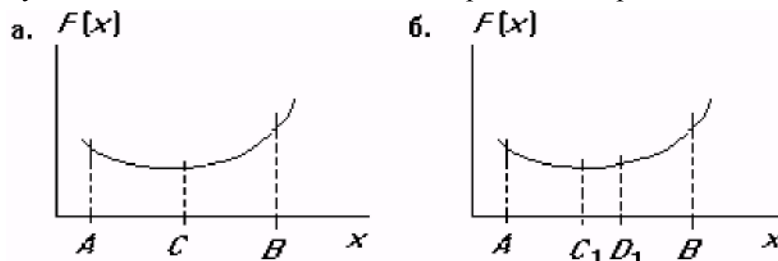


Рис. 4.3. Одновимірна мінімізація: а - дихотомічний розподіл; б - золотий перетин

Нехай заданий відрізок $[A, U]$, на якому мається один мінімум (у загальному випадку непарне число мінімумів).

Відповідно до *методу дихотомічного розподілу* (мал. 4.3,а) відрізок поділяють навпіл і в точках, що відстоять від центра 3 відрізка на величину припустимої погрешності q , розраховують значення цільової функції $F(C+q)$ і $F(C-q)$. Якщо виявиться, що $F(C+q) > F(C-q)$, то мінімум знаходиться на відрізку $[A, 3]$, якщо $F(C+q) < F(C-q)$, те мінімум — на $[3, U]$, якщо $F(C+q) = F(C-q)$ — на $[C-q, C+q]$. Таким чином, на

систем

наступному кроці замість відрізка $[A, Y]$ потрібно досліджувати звужений відрізок $[A, Z]$, $[Z, Y]$ або $[C - q, C + q]$. Кроки повторюються, поки довжина відрізка не зменшиться до величини погрішності q . Таким чином, потрібно не більш N кроків, де N — найближче до $\log((B-A)/q)$ ціле значення, але на кожному кроці цільову функцію варто обчислювати двічі.

По методу золотого січення (мал. 4.3,б) усередині відрізка $[A, Y]$ виділяють дві проміжні крапки Z_1 і D_1 на відстані $s = a$ від його кінцевих крапок, де $L = B - A$ — довжина відрізка. Потім обчислюють значення цільової функції $F(x)$ у точках C_1 і D_1 . Якщо $F(C_1) < F(D_1)$, то мінімум знаходиться на відрізку $[A, D_1]$, якщо $F(C_1) > F(D_1)$, то — на відрізку $[C_1, B]$, якщо $F(C_1) = F(D_1)$ — на відрізку $[C_1, D_1]$. Отже, замість відрізка $[A, Y]$ тепер можна розглядати відрізок $[A, D_1]$, $[Z_1, B]$ або $[C_1, D_1]$, тобто довжина відрізка зменшилася не менш чим у $L/(L-a) = 1/(1-a)$ раз. Якщо підібрати значення a так, що в отриманому відрізку меншої довжини одна з проміжних крапок збіжиться з проміжною точкою від попереднього кроку, тобто у випадку вибору відрізка $[A, D_1]$ точка D_2 збіжиться з точкою C_1 , а у випадку вибору відрізка $[Z_1, B]$ точка Z_2 — із точкою D_1 , то це дозволить скоротити число обчислень цільової функції на всіх кроках (крім першого) у 2 рази.

Умова одержання такого значення a формулюється в такий спосіб $(1-2a)L_k = aL_{k-1}$, відкіля з обліком того, що $L_k/L_{k-1} = 1/(1-a)$, маємо $a = 0,382$. Це значення і називають *золотим перетином*.

Таким чином, потрібно не більш N кроків і $N+1$ обчислення цільової функції, де N можна розрахувати, використовуючи співвідношення $(B-A)/E = (1-a)N$ при заданій погрішності E визначення екстремума.

Відповідно до методу чисел Фібоначі, використовують числа Фібоначі R_i , послідовність яких утвориться за правилом $R_{i+2} = R_{i+1} + R_i$ при $R_0 = R_1 = 1$, тобто ряд чисел Фібоначі має вигляд 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144... Метод аналогічний методу золотого перетину з тією відмінністю, що коефіцієнт a дорівнює відношенню R_{i-2}/R_i , початкове значення i визначається з умови, що R_i повинне бути найменшим числом Фібоначі, що перевищує величину $(B-A)/E$, де E — задана припустима погрішність визначення екстремума. Так, якщо $(B-A)/E = 100$, то початкове значення $i = 12$, оскільки $R_{12} = 144$, і $a = 55/144 = 0,3819$, на наступному кроці буде $a = 34/89 = 0,3820$ і т.д.

По методу поліноміальної апроксимації при апроксимації $F(x)$ квадратичним поліномом

$$P(x) = a_0 + a_1x + a_2x^2 \quad (4.7)$$

вибирають проміжну крапку C і в точках A , Y , Z обчислюють значення цільової функції. Далі вирішують систему з трьох алгебраїчних рівнянь, отриманих підстановкою в (4.7) значень A, Y, Z замість x і обчислених значень функції замість $P(x)$. У результаті стають відомими значення коефіцієнтів a_k у (4.7) і, виходячи з умови $d(x)/dx = 0$, визначають екстремальну крапку \mathcal{E} полінома. Наприклад, якщо точка Z обрана в середині відрізка $[A, Y]$, то $\mathcal{E} = Z + (C-A)(F(A)-F(B)) / (2(F(A)-2F(Z)+F(B)))$.

Методи безумовної оптимізації.

Серед методів нульового порядку в САПР знаходять застосування методи Розенброка, конфігурацій (Хука-Дживса), деформованого багатогранника (Нелдера-Мида), випадкового пошуку. До методів з використанням похідних відносяться методи найшвидшого спуску, сполучених градієнтів, перемінної метрики.

Метод Розенброка є поліпшеним варіантом покоординатного спуску.

Метод покоординатного спуску характеризується вибором напрямків пошуку по черзі уздовж усіх n координатних осей, крок розраховується на основі одномірної оптимізації, критерій закінчення пошуку $|X_k - X_{k+1}| < \epsilon$, де ϵ — задана точність визначення локального екстремуму, n — розмірність простору керованих параметрів. Траєкторія покоординатного спуску для приклада двовимірного простору керованих параметрів показана на мал. 4.4, де X_k — крапки на траєкторії пошуку, x_i — керовані параметри.

Лекція 9. Технічне забезпечення процесу проектування

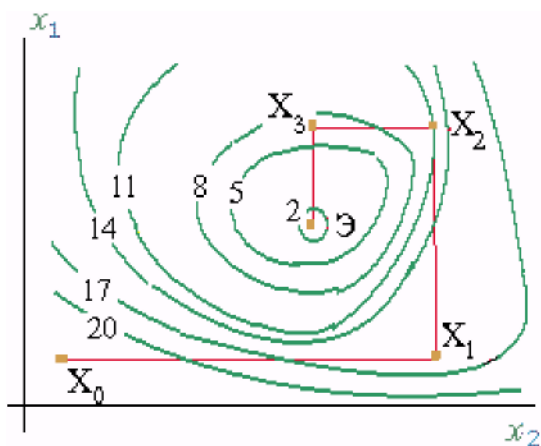


Рис. 4.4. Траєкторія покоординатного спуска

Цільова функція представлена своїми лініями рівного рівня, біля кожної лінії записане відповідне їй значення $F(X)$. Очевидно, що \mathcal{E} є точка мінімуму.

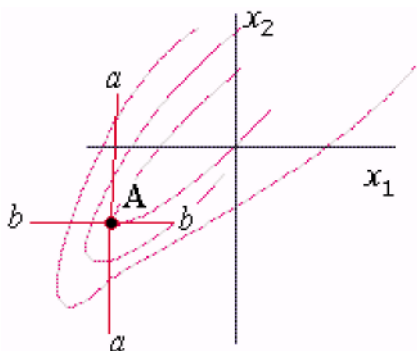


Рис. 4.5. "Застрягання" покоординатного спуску на дні

При використанні методу покоординатного спуска велика імовірність "застревання" пошуку на дні яру удаліні від крапки екстремуму. На мал. 4.5 видно, що після влучення в крапку А, розташовану на дні яру, подальші кроки можливі лише в напрямках aa або bb , але вони приводять до погіршення цільової функції. Отже, пошук припиняється в крапці А.

Примітка. Яром називають частина простору керованих параметрів, у якій спостерігаються слабкі зміни похідних цільової функції по одним напрямкам і значні зміни зі зміною знака — по деяких інших напрямках. Знак похідної міняється в точках, що належать дну яру.

У той же час при сприятливій орієнтації дна яру, а саме при положенні однієї з координатних осей, близькому до паралельності з дном яру, пошук виявляється досить швидким. Ця ситуація показана на мал. 4.6.

Метод Розенброка полягає в такому повороті координатних осей, щоб одна з них виявилася квазіпаралельності дну яру. Такий поворот здійснюють на основі даних, отриманих після серії з n кроків покоординатного спуска. Положення нових осей s_i може бути отримано лінійним перетворенням колишніх осей x_i : вісь S_1 збігається по напрямку з вектором X_{k+n}

систем

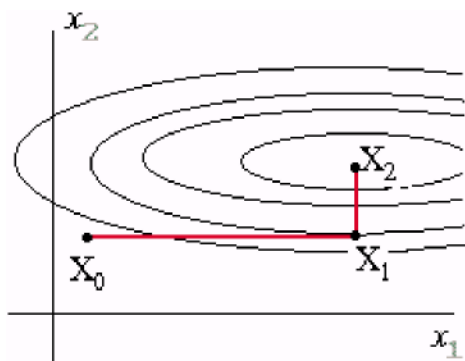


Рис. 4.6. Траєкторія покоординатного спуска при сприятливій орієнтації координатних осей X_k ; інші осі вибирають з умови ортогональності до X_1 і один до одного.

Іншою вдалою модифікацією покоординатного спуска є *метод конфігурацій*. Відповідно до цього методу спочатку виконують звичайну серію з n кроків покоординатного спуска, потім роблять додатковий крок у напрямку вектора $X_k - X_{k-n}$, як показано на мал. 4.7, де додатковий крок виконують у напрямку вектора $X_3 - X_1$, що і приводить у крапку X_4 .

Пошук екстремуму *методом деформованого багатогранника* заснований на побудові багатогранника з $(n + 1)$ вершинами на кожному кроці пошуку, де n — розмірність простору керованих параметрів. На початку пошуку ці вершини вибирають довільно, на наступних кроках вибір підлеглий правилам методу.

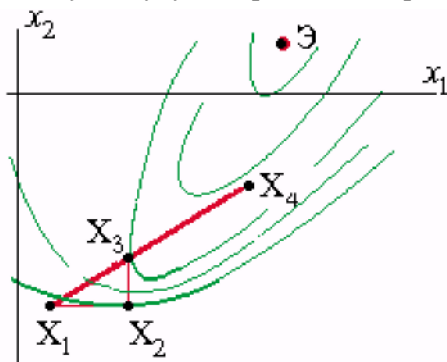


Рис. 4.7. Ілюстрація методу конфігурацій

Ці правила пояснюються мал. 4.8 на прикладі двовимірної задачі оптимізації. Обрано вершини вихідного трикутника: X_1, X_2, X_3 . Нова вершина X_4 знаходиться на промені, проведеному з гіршої вершини X_1 (з вершини з найбільшим значенням цільової функції) через центр ваги ЦТ багатогранника, причому рекомендується X_4 вибирати на відстані d від ЦТ, рівному $|\text{ЦТ} - X_1|$. Нова вершина X_4 заміняє гіршу вершину X_1 . Якщо виявляється, що X_4 має краще значення цільової функції серед вершин багатогранника, то відстань d збільшують. На малюнку саме ця ситуація має місце і збільшення d дає крапку X_5 . У новому багатограннику з вершинами X_2, X_3, X_5 гіршої є вершина X_2 , аналогічно одержують вершину X_6 , потім вершину X_7 і т.д. Якщо нова вершина виявиться гіршої, то в багатограннику потрібно зберегти кращу вершину, а довжини всіх ребер зменшити, наприклад удвічі (стягування багатогранника до кращої вершини). Пошук припиняється при виконанні умови зменшення розмірів багатогранника до деякої межі.

Лекція 9. Технічне забезпечення процесу проектування

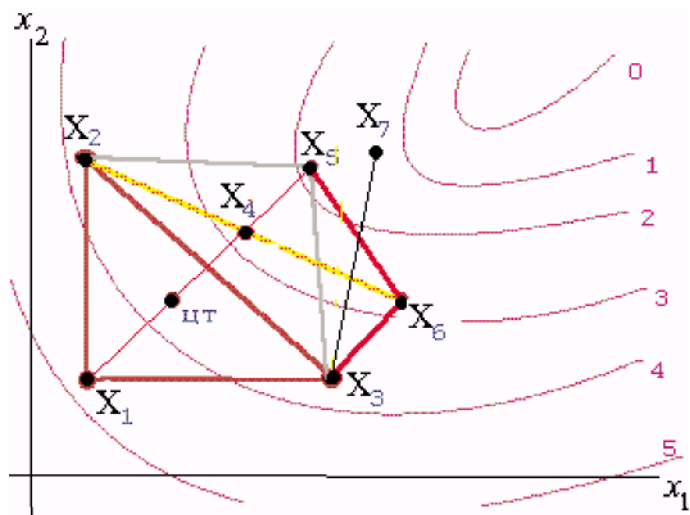


Рис. 4.8. Ілюстрація методу деформованого багатогранника

Випадкові методи пошуку характеризуються тим, що напрямку пошуку g вибирають випадковим чином.

Особливістю *методу найшвидшого спуску* є виконання кроків пошуку в градієнтному напрямку

$$\mathbf{X}_{k+1} = \mathbf{X}_k + h \frac{\mathbf{grad} F(\mathbf{X})}{|\mathbf{grad} F(\mathbf{X})|},$$
 крок h вибирається оптимальним за допомогою одномірної оптимізації.

При використанні методу найшвидшого спуску, як і більшості інших методів, ефективність пошуку істотно знижується в яружних ситуаціях. Траєкторія пошуку здобуває зигзагоподібний вид з повільним просуванням уздовж дна яру у бік екстремуму. Щоб підвищити ефективність градієнтних методів, використовують кілька прийомів.

Один із прийомів, використаний у методі сполучених градієнтів (називаному також методом Флетчера-Ривса), заснований на понятті спряженості векторів. Вектори A і B називають Q -спряженим, якщо $\mathbf{A}^T \mathbf{Q} \mathbf{B} = 0$, де Q — позитивно визначена квадратна матриця того ж порядку, що і розмір N векторів A і B (окремий випадок спряженості — ортогональність векторів, коли Q є одиничною матрицею порядку N), A^T — вектор-рядок, B — вектор-стовпець.

Особливість сполучених напрямків для $Q = \Gamma$, де Γ — матриця Гессе, при в задачах із квадратичною цільовою функцією $F(X)$ полягає в наступному: одномірною мінімізацією $F(X)$ послідовно по N сполучених напрямках дозволяє знайти екстремальну крапку не більш, ніж за N кроків.

Примітка. Матрицею Гессе називають матрицю других часток похідних цільової функції по керованих параметрах.

Підставою для використання пошуку по Γ -сполучених напрямках є те, що для функцій $F(X)$ загального виду може бути застосована квадратична апроксимація, що на практиці виливається у виконання пошуку більш, ніж за N кроків.

Приклад. Пошук екстремуму виконують відповідно до формули

$$\mathbf{X}_i = \mathbf{X}_{i-1} + h \mathbf{S}_i, \quad (4.8)$$

Напрямок \mathbf{S}_{i+1} пошуку на черговому кроці зв'язано з напрямком пошуку \mathbf{S}_i на попередньому кроці співвідношенням

$$\mathbf{S}_{i+1} = -\mathbf{grad} F(\mathbf{X}_i) + w_i \mathbf{S}_i, \quad (4.9) \text{ де } w_i \text{ — коефіцієнт. Крім того, враховують умову спряженості}$$

систем

$$\mathbf{S}_{i+1}^T \mathbf{\Gamma} \mathbf{S}_i = 0 \quad (4.10)$$
 і лінійну апроксимацію $\text{grad}(X)$ в околицях крапки \mathbf{X}_i

$$\text{grad } F(\mathbf{X}_{i+1}) = \text{grad } F(\mathbf{X}_i) + \mathbf{\Gamma}(\mathbf{X}_{i+1} - \mathbf{X}_i). \quad (4.11)$$

Оскільки крок h розраховується виходячи з умови одномірної оптимізації, то, по-перше, справедливе співвідношення

$$\mathbf{S}_i^T \text{grad } F(\mathbf{X}_i) = 0, \quad (4.12)$$

по-друге, маємо

$$\mathbf{X}_i = \mathbf{X}_{i-1} + h w_{i-1} \mathbf{S}_{i-1} - h \text{grad } F(\mathbf{X}_{i-1}),$$

відкіля одержуємо

$$\partial F / \partial h = (\partial F(\mathbf{X}) / \partial \mathbf{X})(\partial \mathbf{X} / \partial h) = \text{grad } F(\mathbf{X}_i) \text{grad } F(\mathbf{X}_{i-1}) = 0. \quad (4.13)$$

Алгоритм пошуку зводиться до застосування формули (4.9), поки не буде виконана умова закінчення обчислень

$$|\text{grad } F(\mathbf{X}_k)| < \varepsilon.$$

Щоб визначити коефіцієнт w_i вирішують систему рівнянь (4.8)-(4.13) шляхом підстановки в (4.10) величин \mathbf{S}_{i+1} з (4.9) і \mathbf{S}_i з (4.8)

$$\begin{aligned} \mathbf{S}_{i+1}^T \mathbf{\Gamma} \mathbf{S}_i &= (w_i \mathbf{S}_i - \text{grad } F(\mathbf{X}_i))^T \mathbf{\Gamma} (\mathbf{X}_i - \mathbf{X}_{i-1}) / h = \\ &= (w_i \mathbf{S}_i - \text{grad } F(\mathbf{X}_i))^T \mathbf{\Gamma} \mathbf{\Gamma}^{-1} (\text{grad } F(\mathbf{X}_i) - \text{grad } F(\mathbf{X}_{i-1})) / h = 0; \end{aligned}$$

або

$$(w_i \mathbf{S}_i - \text{grad } F(\mathbf{X}_i))^T (\text{grad } F(\mathbf{X}_i) - \text{grad } F(\mathbf{X}_{i-1})) = 0,$$

відкіля

$$w_i \mathbf{S}_i^T (\text{grad } F(\mathbf{X}_i) - \text{grad } F(\mathbf{X}_{i-1})) - \text{grad } F(\mathbf{X}_i)^T \text{grad } F(\mathbf{X}_i) + \text{grad } F(\mathbf{X}_i)^T \text{grad } F(\mathbf{X}_{i-1}) = 0$$

і з обліком (4.12) і (4.13)

$$w_i \mathbf{S}_i^T \text{grad } F(\mathbf{X}_{i-1}) + \text{grad } F(\mathbf{X}_i)^T \text{grad } F(\mathbf{X}_i) = 0.$$

Отже,

$$w_i = \text{grad } F(\mathbf{X}_i)^T \text{grad } F(\mathbf{X}_i) / \mathbf{S}_i^T \text{grad } F(\mathbf{X}_{i-1}) \quad (4.14)$$

На першому кроці пошуку вибирають $\mathbf{S}_1 = -\text{grad } F(\mathbf{X}_0)$ і знаходять крапку \mathbf{X}_1 . На другому кроці по формулі (4.14) розраховують w_1 , по формулах (4.9) і (4.8) визначають \mathbf{S}_2 і \mathbf{X}_2 і т.д.

Метод перемінної метрики (інакше метод Девидона-Флетчера-Пауелла) можна розглядати як результат удосконалення методу другого порядку — методу Ньютона. Метод Ньютона заснований на використанні необхідних умов безумовного екстремуму цільової функції $F(X)$

$$\text{grad } F(\mathbf{X}) = 0. \quad (4.15)$$

Виразення (4.15) являє собою систему алгебраїчних рівнянь, для рішення якої можна застосувати відомий чисельний метод, названий методом Ньютона. Корінь системи (4.15) є стаціонарна точка, тобто можливе рішення екстремальної задачі. Метод Ньютона є ітераційним, він заснований на лінеаризації (4.15) в околиці поточної крапки пошуку \mathbf{X}_k

$$\text{grad } F(\mathbf{X}) = \text{grad } F(\mathbf{X}_k) + \mathbf{\Gamma}(\mathbf{X} - \mathbf{X}_k) = 0. \quad (4.16)$$

Виразення (4.16) — це система лінійних алгебраїчних рівнянь. Її корінь є чергове наближення \mathbf{X}_{k+1} до рішення $\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{\Gamma}^{-1}(\mathbf{X}_k) \text{grad } F(\mathbf{X}_k)$.

Якщо процес сходиться, то рішення досягається за мале число ітерацій, закінченням яких служить виконання умови:

$$|\mathbf{X}_{k+1} - \mathbf{X}_k| < \varepsilon.$$

Головний недолік методу — висока трудомісткість обчислення і звертання матриці $\mathbf{\Gamma}$, до того ж її обчислення чисельним диференціюванням супроводжується помітними погрішностями, що знижує швидкість збіжності.

Лекція 9. Технічне забезпечення процесу проектування

У методі перемінної метрики замість зворотної матриці, що обчислюється важко, Гессе використовують якусь більш матрицю, що легко обчислюється, \mathbf{N} , тобто

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \mathbf{N} \mathbf{grad} F(\mathbf{X}_k).$$

Уведемо позначення:

$$d\mathbf{g}_k = \mathbf{grad} F(\mathbf{X}_k) - \mathbf{grad} F(\mathbf{X}_{k-1}); \quad d\mathbf{X}_k = \mathbf{X}_k - \mathbf{X}_{k-1};$$

\mathbf{E} — одинична матриця. Початкове значення матриці $\mathbf{N}_0 = \mathbf{E}$. Матрицю \mathbf{N} коректують на кожному кроці, тобто

$$\mathbf{N}_{k+1} = \mathbf{N}_k + \mathbf{A}_k - \mathbf{B}_k,$$

де

$$\mathbf{A}_k = d\mathbf{X}_k d\mathbf{X}_k^T / (d\mathbf{X}_k^T d\mathbf{g}_k),$$

$$\mathbf{B}_k = \mathbf{N}_k d\mathbf{g}_k d\mathbf{g}_k^T \mathbf{N}_k^T / (d\mathbf{g}_k^T \mathbf{N}_k d\mathbf{g}_k).$$

Тому

$$\mathbf{N}_{k+1} = \mathbf{E} + \sum_{i=0}^k \mathbf{A}_i - \sum_{i=0}^k \mathbf{B}_i.$$

Можна показати, що \mathbf{A}_i прагне до \mathbf{I}^{-1} , $\mathbf{B}_i \rightarrow \mathbf{E}$ при $k \rightarrow n$, де n — розмірність простору керування параметрів. Опісля n кроків, потрібно знову починати з $\mathbf{N}_{n+1} = \mathbf{E}$.

Необхідні умови екстремуму.

У задачах безумовної оптимізації необхідні умови являють собою рівність нулеві градієнта цільової функції

$$\mathbf{grad} F(\mathbf{X}) = \mathbf{0}.$$

У загальній задачі математичного програмування (4.1) необхідні умови екстремуму, називані умовами Куна-Таккера, формулюються в такий спосіб:

Для того щоб точка \mathcal{E} була екстремальною точкою опуклої задачі математичного програмування (ЗМП), необхідна наявність ненегативних коефіцієнтів u_i , таких, що

$$u_i \varphi_i(\mathcal{E}) = 0, \quad i = 1, 2, \dots, m; \quad (4.17)$$

і при цьому дотримувалися обмеження задачі, а також виконувалася умова

$$\mathbf{grad} F(\mathcal{E}) + \sum_{i=1}^m u_i \mathbf{grad} \varphi_i(\mathcal{E}) + \sum_{j=1}^L a_j \psi_j(\mathcal{E}) = \mathbf{0}, \quad (4.18)$$

де m — число обмежень типу нерівностей, L — те ж рівностей, коефіцієнти $a_j > 0$

За приведеним абстрактним формулюванням умов ховається досить геометричний зміст, що розуміється просто. Дійсно, розглянемо спочатку випадок з обмеженнями тільки типу нерівностей. Якщо максимум знаходиться усередині припустимої області R , то, вибираючи всі $u_i = 0$, домагаємося виконання (4.17); якщо ж точка максимуму \mathcal{E} лежить на границі області R , те, як видно з лівої частини мал. 4.9, цю крапку завжди відповідним підбором ненегативних u_i можна помістити усередину оболонки, натягнутої на градієнти цільової функції $F(X)$ і функцій-обмежень $\varphi_i(X)$.

систем

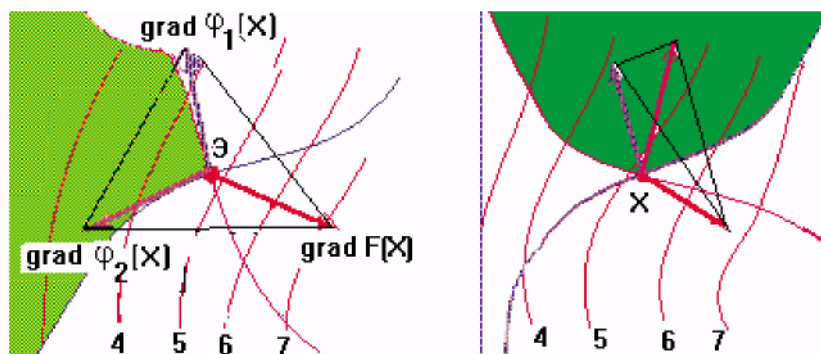


Рис. 4.9. До пояснення умов Куна-Таккера

Навпаки, якщо точка не є екстремальною, те (4.17) не можна виконати при будь-якому виборі позитивних коефіцієнтів u_i (див. праву частину мал. 4.9, де розглянута точка X лежить поза опуклою оболонкою, натягнутою на градієнти). Облік обмежень типу рівностей очевидний, якщо додається остання з зазначених у (4.18) сума.

Методи пошуку умовних екстремумів.

Широко відомий метод *множників Лагранжа*, орієнтований на пошук екстремуму при наявності обмежень типу рівностей $\psi(X) = 0$, тобто на рішення задачі

$$\text{extr}_{X \in \mathbf{R}} F(\mathbf{X}), \quad (4.19)$$

де $\mathbf{R} = \{ \mathbf{X} \mid \psi(\mathbf{X}) = 0 \}$.

Суть методу полягає в перетворенні задачі умовної оптимізації (4.19) у задачу безумовної оптимізації за допомогою утворення нової цільової функції

$$\Phi(\mathbf{X}, \mathbf{V}) = F(\mathbf{X}) + \sum_{i=1}^L \lambda_i \psi_i(\mathbf{X}),$$

де $\mathbf{\Lambda} = (\lambda_1, \lambda_2, \lambda_3 \dots \lambda_h)$ — вектор множників Лагранжа, L — число обмежень.

Необхідні умови екстремуму функції $\Phi(\mathbf{X})$:

$$\partial \Phi(\mathbf{X}, \mathbf{\Lambda}) / \partial \mathbf{X} = \partial F(\mathbf{X}) / \partial \mathbf{X} + \sum_{i=1}^L \lambda_i \partial \psi_i(\mathbf{X}) / \partial \mathbf{X} = 0; \quad \partial \Phi(\mathbf{X}, \mathbf{\Lambda}) / \partial \mathbf{\Lambda} = \psi(\mathbf{X}) = 0. \quad (4.20)$$

Система (4.20) містить $n+L$ алгебраїчних рівнянь, де n — розмірність простору керованих параметрів, її рішення дає шукані координати екстремальної крапки і значення множників Лагранжа. Однак при чисельному рішенні (4.20), що має місце при використанні алгоритмічних моделей, виникають ті ж труднощі, що й у методі Ньютона. Тому в САПР основними методами рішення ЗМП є методи штрафних функцій і проекції градієнта.

Основна ідея *методів штрафних функцій* — перетворення задачі умовної оптимізації в задачу безумовної оптимізації шляхом формування нової цільової функції $\Phi(X)$ введенням у вихідну цільову функцію $F(X)$ спеціальним образом обраної функції штрафу $S(X)$:

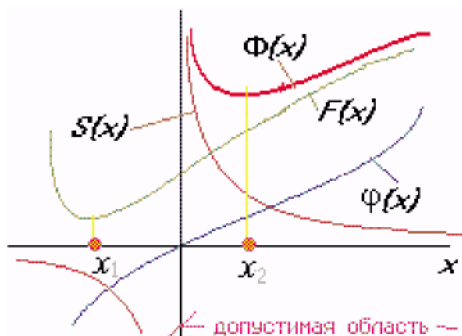
$$\Phi(\mathbf{X}) = F(\mathbf{X}) + rS(\mathbf{X}),$$

де r — множник, значення якого можна змінювати в процесі оптимізації.

Серед методів штрафних функцій розрізняють методи внутрішньої і зовнішньої крапки. Відповідно до методів внутрішньої крапки (інакше називаним методами *бар'єрних функцій*) вихідну для

Лекція 9. Технічне забезпечення процесу проектування

пошуку крапку можна вибирати тільки усередині припустимої області, а для методів зовнішньої крапки як усередині, так і поза припустимою областю (важливо лише, щоб у ній функції цільова й обмежень були б визначені). Ситуація появи бар'єра в цільовій функції $\Phi(x)$ і співвідношення між умовним у крапці x_2 і безумовним у крапці x_1 мінімумами $F(x)$ у найпростішому одномірному випадку ілюструється мал. 4.10.



Приклади штрафних функцій:

- 1) для методу внутрішньої крапки при обмеженнях $\varphi_i(\mathbf{X}) > \text{ПРО}$

$$S(\mathbf{X}) = \sum_{i=1}^m (1 / \varphi_i(\mathbf{X})),$$

де m — число обмежень типу нерівностей;

- 2) для методу зовнішньої крапки при таких же обмеженнях

Рис. 4.10. Пояснення методу штрафних функцій

$$S(\mathbf{X}) = \sum_{i=1}^m (\min\{0, \varphi_i(\mathbf{X})\})^2 —$$

тут штраф зводиться до включення у $\Phi(\mathbf{X})$ суми квадратів активних (тобто порушених) обмежень;

- 3) у випадку обмежень типу рівностей $\psi_i(\mathbf{X}) = 0$

$$S(\mathbf{X}) = \sum_{i=1}^L (\psi_i(\mathbf{X}))^2.$$

Чим більше коефіцієнт r , тим точніше рішення задачі, однак при великих r може погіршуватися її обумовленість. Тому на початку пошуку звичайно вибирають помірні значення r , збільшуючи них у околицях екстремуму.

Основний варіант *методу проекції градієнта* орієнтований на задачі математичного програмування з обмеженнями типу рівностей.

Пошук при виконанні обмежень здійснюється в підпросторі $(n-m)$ вимірів, де n — число керованих параметрів, m — число обмежень, при цьому рух здійснюється в напрямку проекції градієнта цільової функції $F(\mathbf{X})$ на гіперплощину, дотичну до гіперповерхні обмежень (точніше до гіперповерхні перетинання гіперповерхонь обмежень).

Пошук мінімуму починають зі спуску з вихідної точки на гіперповерхню обмежень. Далі виконують крок у зазначеному вище напрямку (крок уздовж гіперповерхні обмежень). Оскільки цей крок може привести до помітного порушення обмежень, знову повторюють спуск на гіперповерхність обмежень і т.д. Іншими словами, пошук полягає у виконанні пар кроків, кожна пара включає спуск на гіперповерхню обмежень і рух уздовж гіперповерхні обмежень.

систем

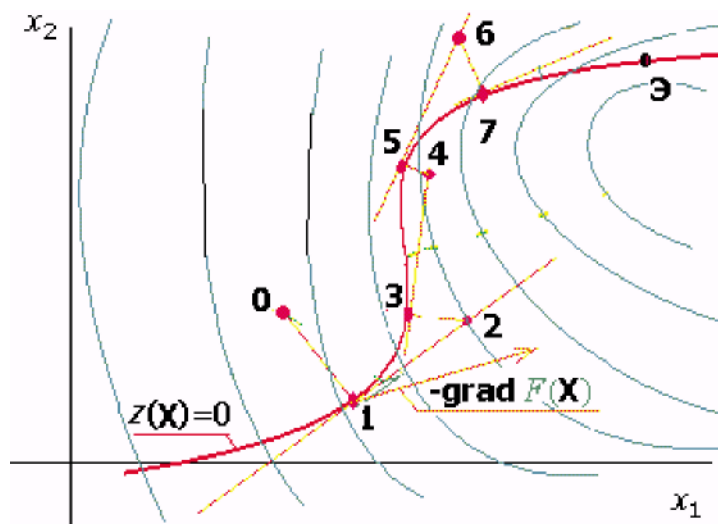


Рис. 4.11. Траєкторія пошуку відповідно до методу проєкції градієнта: Э - умовний екстремум; 0,1,2,3- крапки на траєкторії пошуку

Ідею методу легко пояснити для випадку пошуку в двовимірному просторі при одному обмеженні $\psi(\mathbf{X}) = 0$. На мал. 4.11 це обмеження представлено жирною лінією, а цільова функція — сукупністю більш тонких ліній рівного рівня. Спуск звичайно здійснюють по нормалі до гіперповерхні обмежень (у даному випадку до лінії обмеження). Умова закінчення пошуку засновано на зіставленні значень цільової функції в двох послідовних точках, одержуваних після спуска на гіперповерхню обмежень.

Розглянемо питання, що стосується одержання аналітичних виражень для напрямків спуска і руху уздовж гіперповерхні обмежень.

Спуск. Необхідно з поточної крапки пошуку В потрапити в крапку А, що є найближчою до У точкою на гіперповерхні обмежень, тобто вирішити задачу

$$\min |\mathbf{B}-\mathbf{A}|$$

за умови $\psi(\mathbf{X})=0$, що після лінеаризації в околицях крапки В має вигляд

$$\psi(\mathbf{B}) + (\mathbf{grad} \psi(\mathbf{B}))^T(\mathbf{A}-\mathbf{B}) = 0.$$

Використовуючи метод множників Лагранжа, позначаючи $\mathbf{A}-\mathbf{B}=\mathbf{U}$ і з огляду на, що мінімізація відстані рівнозначна мінімізації скалярного добутку \mathbf{U} на \mathbf{U} , запишемо

$$\Phi(\mathbf{A}) = \mathbf{U}^T\mathbf{U} + \lambda (\psi(\mathbf{B})+(\mathbf{grad} \psi(\mathbf{B}))^T\mathbf{U});$$

$$\partial\Phi/\partial\mathbf{A} = 2\mathbf{U} + \lambda (\mathbf{grad} \psi(\mathbf{B})) = 0; \quad (4.21)$$

$$\partial\Phi/\partial\lambda = \psi(\mathbf{B}) + (\mathbf{grad} \psi(\mathbf{B}))^T\mathbf{U} = 0. \quad (4.22)$$

Тоді з (4.21) одержуємо вираження

$$\mathbf{U} = -0,5\lambda (\mathbf{grad} \psi(\mathbf{B})),$$

підставляючи його в (4.22), маємо

$$\psi(\mathbf{B}) - 0,5\lambda (\mathbf{grad} \psi(\mathbf{B}))^T \mathbf{grad} \psi(\mathbf{B}) = 0;$$

відкіля

$$\lambda = (0,5(\mathbf{grad} \psi(\mathbf{B}))^T \mathbf{grad} \psi(\mathbf{B}))^{-1}\psi(\mathbf{B}).$$

і остаточно, підставляючи λ в (4.21), знаходимо

$$\mathbf{U} = -\mathbf{grad} \psi(\mathbf{B})(\mathbf{grad} \psi(\mathbf{B}))^T \mathbf{grad} \psi(\mathbf{B})^{-1}\psi(\mathbf{B}).$$

Рух уздовж гіперповерхні обмежень. Крок у гіперплощині D, дотичної до гіперповерхні обмежень, варто зробити в напрямку вектора S, на якому цільова функція зменшується в найбільшій мері при

Лекція 9. Технічне забезпечення процесу проектування заданому кроці h . Зменшення цільової функції при переході з точки A в нову точку C підраховують, використовуючи формулу лінеаризації $F(X)$ в околицях точки A :

$$F(C) - F(A) = h(\text{grad } F(A))^T S,$$

де $\text{grad } F(A)^T S$ — збільшення $F(X)$, яких потрібно мінімізувати, варіюючи напрямку S

$$\min F(C) = \min ((\text{grad } F(A))^T S), \quad (4.23) \text{ де варіація } S \text{ здійснюється в межах гіперплощини}$$

D ; $\text{grad } \psi(A)$ і S — ортогональні вектори.

Отже, мінімізацію (4.23) необхідно виконувати при обмеженнях

$$(\text{grad } \psi(A))^T S = 0,$$

$$S^T S = 1.$$

Останнє обмеження говорить про те, що при пошуку напрямку руху, вектор S повинний лише вказувати цей напрям, тобто його довжина несуттєва (нехай S — одиничний вектор). Для рішення (4.23) використовуємо метод множників Лагранжа

$$\Phi(S, \lambda, q) = (\text{grad } F(A))^T S + \lambda (\text{grad } \psi(A))^T S + q(S^T S - 1), \quad (4.24)$$

де λ і q — множники Лагранжа;

$$\partial \Phi / \partial S = \text{grad } F(A) + \lambda \text{grad } \psi(A) + qS = 0;$$

$$\partial \Phi / \partial \lambda = (\text{grad } \psi(A))^T S = 0; \quad (4.25)$$

$$\partial \Phi / \partial q = S^T S - 1 = 0. \quad (4.26) \text{ З (4.24) випливає, що}$$

$$S = -(\text{grad } F(A) + \lambda \text{grad } \psi(A)) / q;$$

підставляючи S у (4.25), одержуємо

$$(\text{grad } \psi(A))^T \text{grad } F(A) + \lambda (\text{grad } \psi(A))^T \text{grad } \psi(A) = 0,$$

звідкіля

$$\begin{aligned} \lambda &= - [(\text{grad } \psi(A))^T \text{grad } \psi(A)]^{-1} (\text{grad } \psi(A))^T \text{grad } F(A), \quad S = \\ &= - \{ \text{grad } F(A) - \text{grad } \psi(A) [(\text{grad } \psi(A))^T \text{grad } \psi(A)]^{-1} (\text{grad } \psi(A))^T \text{grad } F(A) \} / q = \\ &= - \{ E - \text{grad } \psi(A) [(\text{grad } \psi(A))^T \text{grad } \psi(A)]^{-1} (\text{grad } \psi(A))^T \} \text{grad } F(A) / q. \end{aligned} \quad (4.27)$$

Таким чином, матриця

$$P = E - \text{grad } \psi(A) [(\text{grad } \psi(A))^T \text{grad } \psi(A)]^{-1} \text{grad } \psi(A)^T$$

являє собою матрицю, що проектує, а вектор S , розрахований по (4.27), — проекцію градієнта $\text{grad}(A)$ на гіперповерхню обмежень.

Часткою случаємо застосування методу проекції градієнта є задачі оптимізації з максимумним критерієм. Дійсно, для пошуку екстремуму функції мінімуму

$$\max_{\mathbf{X}} \min_j Z_j(\mathbf{X}),$$

де Z_j — нормована величина у j -го вихідного параметра u_j , зручно застосовувати метод проекції градієнта. Як обмеження задачі у вихідній постановці фігурують тільки прямі обмеження

$$x_{\max i} > x_i > x_{\min i}.$$

Тут $x_{\max i}$ і $x_{\min i}$ — граничні значення припустимого діапазону варіювання параметра x_i . У процесі пошуку, якщо мінімальної є функція $Z_q(X)$ і траєкторія пошуку перетинає гребінь

$$Z_q(\mathbf{X}) - Z_k(\mathbf{X}) = 0, \quad (4.28)$$

те пошук продовжується в напрямку проекції градієнта функції $Z_q(X)$ на гіперповерхню гребеня (4.28).

22. Математичне забезпечення структурного синтезу в АПСОС

Постановка задач структурного синтезу

Процедури синтезу проектних рішень.

Прийняття проектних рішень охоплює широке коло задач і процедур — від вибору варіантів у кінцевих і доступних для огляду множинах до задач творчого характеру, що не мають формальних способів рішення.

Відповідно в САПР застосовують як засобу формального синтезу проектних рішень, виконуваного в автоматичному режимі, так і допоміжні засоби, що сприяють виконанню синтезу проектних рішень в інтерактивному режимі. До допоміжних засобів відносяться бази типових проектних рішень, системи навчання проектуванню, програмно-методичні комплекси верифікації проектних рішень, уніфіковані мови опису ТЗ і результатів проектування.

Задачі синтезу структур проєктованих об'єктів відносяться до найбільше важко формалізуєм. Існує ряд загальних підходів до постановки цих задач, однак практична реалізація більшості з них неочевидна. Тому маються лише "острівці" автоматичного виконання процедур синтезу серед "моря" проблем, що чекають автоматизації.

Саме з цієї причини структурний синтез, як правило, виконують в інтерактивному режимі при вирішальній ролі інженера-розроблювача, а ЕОМ відіграє допоміжну роль: надання необхідних довідкових даних, фіксація й оцінка проміжних і остаточних результатів.

Однак у ряді додатків маються і приклади успішної автоматизації структурного синтезу в ряді додатків; серед них заслуговують згадування в першу чергу задачі конструкторського проектування друкованих плат і кристалів БІС, логічного синтезу комбінаційних схем цифрової автоматики й обчислювальної техніки, синтезу технологічних процесів і керуючих програм для механічної обробки в машинобудуванні і деякі інші.

Структурний синтез полягає в перетворенні описів проєктованого об'єкта: вихідний опис містить інформацію про вимоги до властивостей об'єкта, про умови його функціонування, обмеження на елементний склад і т.п., а результуючий опис повинний містити зведення про *структуру*, тобто про склад елементів і способи їхнього з'єднання і взаємодії.

Постановки і методи рішення задач структурного синтезу в зв'язку з труднощами формалізації не досягли ступеня узагальнення і деталізації, властиві математичному забезпеченню процедур аналізу. Досягнутий ступінь узагальнення виражається у встановленні типової послідовності дій і використовуваних видів описів при їхніх перетвореннях у САПР. Вихідний опис, як правило, являє собою ТЗ на проектування, по ньому складають опис на деякій формальній мові, що є вхідною мовою використовуваних підсистем САПР. Потім виконують перетворення описів, і одержуваного підсумкове для даного етапу опис документують — представляють у виді твердої копії або файлу у відповідному форматі для передачі на наступний етап.

Важливе значення для розвитку підсистем синтезу в САПР мають розробка й уніфікація мов представлення описів (специфікацій). Кожна мова, підтримуючи обрану методику прийняття рішень, формує в користувачів САПР — розроблювачів технічних об'єктів визначений стиль мислення; особливості мов безпосередньо впливають на особливості правил перетворення специфікацій. Прикладами уніфікованих мов опису проектних рішень є мова VHDL для радіоелектроніки, вона сполучить у собі

Лекція 9. Технічне забезпечення процесу проектування засобу для функціональних, поведінкових і структурних описів, або мова Express — універсальна мова специфікацій для представлення й обміну інформацією в комп'ютерних середовищах.

Задача прийняття рішень.

Мається ряд підходів для узагальненого опису задач прийняття проектних рішень у процесі структурного синтезу.

Задачу прийняття рішень (ЗПР) формують у такий спосіб:

$$\text{ЗПР} = \langle A, \text{ДО}, \text{Мод}, \Pi \rangle,$$

де A — множин альтернатив проектного рішення, $\text{ДО} = (\text{ДО}_1, \text{ДО}_2, \dots, \text{ДО}_m)$ — множин критеріїв (вихідних параметрів), по яких оцінюється відповідність альтернативи поставленим цілям; Мод: $A \rightarrow \text{ДО}$ — модель, що дозволяє для кожної альтернативи розрахувати вектор критеріїв, Π — вирішальне правило для вибору найбільш підходящої альтернативи в багатокритеріальній ситуації.

У свою чергу, кожній альтернативі конкретного додатка можна поставити у відповідність значення упорядкованої множини (набору) атрибутів $X = \langle x_1, x_2, \dots, x_n \rangle$, що характеризує властивості альтернативи. При цьому x_i може бути величиною типу real, integer, boolean, string (в останньому випадку величину називають предметною або лінгвістичною). Множин X називають записом (у теорії баз даних), фреймом (у штучному інтелекті) або хромосомою (у генетичних алгоритмах). Модель Мод називають структурно-критеріальна, якщо серед x_i маються параметри, що характеризують структуру моделювання об'єкта.

Основними проблемами в ЗПР є:

- компактне представлення множині варіантів (альтернатив);
- побудова моделі синтезованого пристрою, у тому числі вибір ступеня абстрагування для оцінки значень критеріїв;
- формулювання переваг у багатокритеріальних ситуаціях (тобто перетворення векторного критерію K в скалярну цільову функцію);
- установлення порядку (переваг) між альтернативами під час відсутності кількісної оцінки цільової функції (що звичайно є наслідком не кількісного характеру всіх або частини критеріїв);
- вибір методу пошуку оптимального варіанта (скорочення перебору варіантів).

Властивим проектним задачам невизначеність і нечіткість вихідних даних, а іноді і моделей, диктують використання спеціальних методів кількісного формулювання вихідних не кількісних даних і відносин. Ці спеціальні методи або відносяться до області побудови вимірювальних шкал, або є предметом теорії нечітких множиней.

Вимірювальні шкали можуть бути:

- 1) абсолютними;
- 2) номінальними (класифікаційними), значення шкали представляють класи еквівалентності, прикладом може служити шкала квітів; такі шкали відповідають величинам не кількісного характеру;
- 3) порядковими, якщо між об'єктами A і B встановлено одне з наступних відносин: простого порядку, що говорить, що якщо A краще B , то B гірше A , і дотримується транзитивність; або слабкого порядку, тобто або A не гірше B , або A не краще B ; або часткового порядку. Для формування цільової функції $F(X)$ виробляється оцифровка порядкової шкали, тобто при мінімізації, якщо A перед шанобливіше B , то $F(X_a) < F(X_b)$, де X_a і X_b — множині атрибутів об'єктів A і B відповідно;
- 4) інтервальними, що відбивають кількісні відносини інтервалів: шкала єдина з точністю до лінійних перетворень, тобто $y = ax + b$, $a > 0$, $-\infty < b < \infty$, або $y = ax$ при $a \geq 0$, або $v = x + b$.

У більшості випадків структурного синтезу математична модель у виді алгоритму, що дозволяє по заданій множині X і заданій структурі об'єкта розрахувати вектор критеріїв ДО, виявляється відомою. Наприклад, такі моделі виходять автоматично в програмах аналізу типу Spice, Adams або ПА-9 для об'єктів, досліджуваних на макрорівні. Однак у ряді інших випадків такі моделі невідомі в силу недостатньої вивченості процесів і їхніх взаємозв'язків у досліджуваному середовищі, але відома сукупність результатів спостережень або експериментальних досліджень. Тоді для одержання моделей використовують спеціальні методи ідентифікації й апроксимації (моделі, отримані подібним шляхом іноді називають феноменологічні).

систем

Серед методів формування моделей по експериментальним даним найбільш відомі *методи планування експериментів*. Не менш популярним стає підхід, заснований на використанні *штучних нейронних мереж*.

Якщо ж математична модель $X \rightarrow$ До залишається невідомою, то намагаються використовувати підхід на базі *систем штучного інтелекту (експертних систем)*.

Можливості практичного рішення задач *дискретного математичного програмування* (ДМП) вивчаються в теорії складності задач вибору, де показано, що задачі навіть помірного розміру, що відносяться до класу NP-повних задач, у загальному випадку вдається вирішувати тільки приблизно.

Тому більшість практичних задач структурного синтезу вирішують за допомогою наближених (евристичних) методів. Це методи, що використовують специфічні особливості того або іншого класу задач і не гарантуючі одержання оптимального рішення. Часто вони приводять до результатів, близьким до оптимальних, при прийнятних витратах обчислювальних ресурсів.

Якщо всі керовані параметри альтернатив, що позначаються у виді множині X , є кількісними оцінками, то використовують *наближені методи* оптимізації. Якщо в X входять також параметри не кількісного характеру і простір X , то перспективними є *еволюційні методи* обчислень, серед яких найбільш розвинуті *генетичні методи*. Нарешті, під час відсутності обґрунтованих моделей Мод їх створюють, ґрунтуючись на експертних знаннях у виді деякої системи штучного інтелекту.

Представлення множини альтернатив.

Рішенню проблем упорядкування й описи множині альтернатив і зв'язків між ними в конкретних додатках присвячена спеціальна область знання, що за аналогією з наукою опису множин тварин і рослин у біології можна назвати *систематикою*.

Найпростіший спосіб завдання множині A — явне перерахування всіх альтернатив. Семантика і форма опису альтернатив істотно залежать від додатка. Для представлення таких описів у пам'яті ЕОМ і доступу до них використовують *інформаційно-пошукові системи* (ІПС). Кожній альтернативі в ІПС відповідає пошуковий образ, що складається зі значень атрибутів x_i і ключових слів вербальних характеристик.

Явне перерахування альтернатив при представленні множині альтернатив можливо лише при малій потужності A . Тому в більшості випадків використовують неявний опис A у виді способу (алгоритму або набору правил P) синтезу проектних рішень з обмеженого набору елементів \mathcal{E} . Тому тут $A = \langle P, \mathcal{E} \rangle$, а типовий процес синтезу проектних рішень складається з наступних етапів:

- 1) формування альтернативи A_i (це може бути вибір з бази даних ІПС по сформованому пошуковому розпорядженню або генерація з \mathcal{E} в відповідності з правилами P);
- 2) оцінка альтернативи за результатами моделювання за допомогою моделі Мод;
- 3) ухвалення рішення (виконується ЛПР — особою, що приймає рішення, або автоматично) щодо переходу до наступної альтернативи або припинення пошуку.

Для опису множин P і \mathcal{E} використовують наступні підходи.

Морфологічні таблиці й альтернативні І-або-дерева.

Представлення знань в *інтелектуальних системах* — фрейми, семантичні мережі, продукції.

Генетичні методи.

Бази *фізичних ефектів* і *евристичних прийомів*, застосовувані при рішенні задач винахідлового характеру.

Морфологічні таблиці.

Морфологічна таблиця (M) являє собою узагальнену структуру у виді множині функцій, виконуваних компонентами синтезованих об'єктів розглянутого класу, і підмножин способів їхньої реалізації. Кожній функції можна поставити у відповідність один рядок таблиці, кожному способу її реалізації — одну клітку в цьому рядку. Отже, у морфологічних таблицях елемент M_{ij} відповідає j -й варіанту реалізації i -ї функції в класі технічних об'єктів, описуваному матрицею M .

Іншими словами, множин альтернатив можна представити у виді відносини M , названого морфологічною таблицею

$$M = \langle X, R \rangle,$$

де X — множин властивостей (характеристик або функцій), властивим об'єктам розглянутого типу, n — число цих властивостей, $R = \langle R_1, R_2, \dots, R_n \rangle$, R_i — множин значень (способів реалізації) i -го властивості, потужність цієї множині далі позначена N_i . При цьому власне множин альтернатив A представлено композицією множин R_i , тобто кожна альтернатива включає по одному елементі (значенню) з кожного рядка морфологічної таблиці. Очевидно, що загальне число альтернатив k , що представляються морфологічною таблицею, дорівнює

$$k = \prod_{i=1}^n N_i.$$

Морфологічні таблиці звичайно вважають засобом неавтоматизованого синтезу, що допомагає людині переглядати компактно представлені альтернативи, переборювати психологічну інерцію. Останнє зв'язано з тим, що увага ЛПР звертається на варіанти, що без морфологічної таблиці залишалися б поза його полю зору.

Власне таблиця M не містить зведень про спосіб синтезу. Однак на базі M можлива побудова методів синтезу з елементами алгоритмізації. У таких методах вводиться метризація морфологічного простору. Морфологічний простір складають можливі закінчені структури, приймається, що відстань між структурами S_1 і S_2 є число незбіжних елементів (кожна клітка M є один елемент). Тому можна говорити про околиці рішень. Далі виходять із припущення про компактність "гарних" рішень, що дозволяє замість повного перебору обмежуватися перебором у малій околиці поточної крапки пошуку. Таким чином, гіпотеза про "компактність" і метризація простору рішень фактично приводять до побудови математичної моделі, до якого можна застосувати методи дискретної оптимізації, наприклад локальні методи.

До недоліків M відносяться неврахування заборонених сполучень елементів у закінчених структурах і відображення складу елементів у структурах без конкретизації їхніх зв'язків. Крім того, морфологічні таблиці будують у припущенні, що множині R_j взаємно незалежні, тобто склад способів реалізації i -ї функції не міняється при зміні значень інших функцій. Очевидно, що припущення про взаємну незалежність множин R_i виправдано лише в порівняно простих структурах. Останній недолік усувається шляхом узагальнення методу морфологічних таблиць при використанні методу альтернативних (І-АБО) графів.

Альтернативні графи.

Будь-яку морфологічну таблицю можна представити у виді дерева (мал. 4.12). На малюнку функції представлені вершинами I (темні кружки), значення функцій — вершинами АБО (світлі кружки). Очевидно, що таблиця представляє множин однотипних об'єктів, оскільки усі вони характеризуються тою самою множиною функцій.

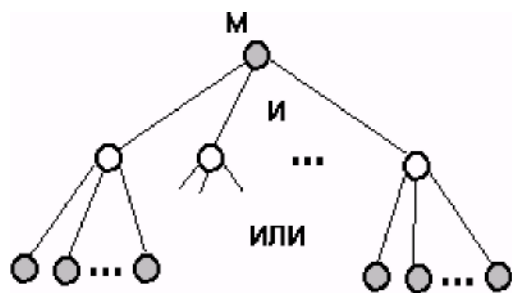


Рис. 4.12. Дерево, що відповідає морфологічній таблиці

Для різнотипних об'єктів застосовують багаторушні альтернативні графи. Наприклад, на мал. 4.13 показаний двох'ярусний граф, у якому для різних типів об'єктів передбачені різні підмножини функцій.

Якщо допустити деяку надмірність при зображенні І-АБО- графа, то його можна перетворити в І-АБО-дерево, що веде до визначених зручностей.

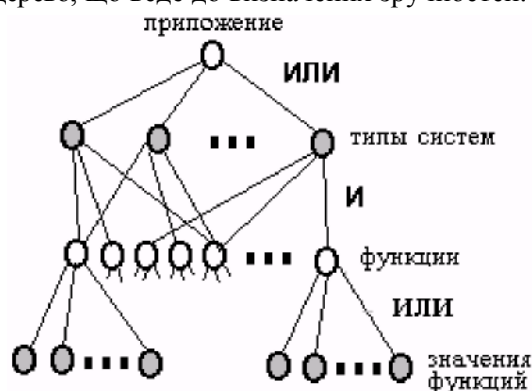


Рис. 4.13. І-або-граф

Очевидно, що І-або-дерево можна представити як сукупність морфологічних таблиць. Кожна І вершина дерева відповідає приватній морфологічній таблиці, тобто множині функцій так, що і-я вихідна галузь відображає і-ю функцію.

Кожна АБО вершина, інцидентна і-й галузі, відповідає множині варіантів реалізації і-й функції, при цьому j-я вихідна з АБО вершини галузь відображає j-й варіант реалізації.

Алгоритмізація синтезу на базі І-або-дерев вимагає введення правил вибору альтернатив у кожній вершині АБО. Ці правила найчастіше мають евристичний характер, зв'язані з вимогами ТЗ, можуть відбивати заборони на сполучення визначених компонентів структур. Труднощі ефективного рішення задачі істотно зростають при наявності обмежень, типовими серед яких є обмеження на сумісність способів реалізації різних функцій, тобто обмеження виду

$$C_{ij} \text{ and } C_{pq} = \text{false}, \quad (4.29)$$

де $C_{ij} = \text{true}$, якщо в оцінюваний варіант ввійшов елемент, \mathcal{E}_{ij} інакше. $C_{ij} = \text{false}$ Умова (4.29) означає, що в припустиму структуру не можуть входити одночасно елементи \mathcal{E}_{ij} і \mathcal{E}_{pq} . Сукупність обмежень типу (4.29) можна представити як систему логічних рівнянь з невідомими C_{ij} . Тоді задачу синтезу можна вирішувати еволюційними методами, якщо попередньо або одночасно з нею вирішувати систему логічних рівнянь (задачу про виконимості).

Вирахування.

Очевидно, що в більшості випадків структурного синтезу замість нереалізованого явного представлення всієї множини проектних рішень задають множин елементів і сукупність правил об'єднання цих елементів у припустимі структури (проектні рішення).

Ці множини елементів і правил часто представляють у виді *формальної системи (вирахування)*, тобто задача синтезу має вигляд

$$ЗС = \langle \mathcal{E}; \mathcal{HT}; \mathcal{AK}; \mathcal{P} \rangle,$$

де \mathcal{E} — алфавіт вирахування (алфавіт представлений базовими елементами, з яких синтезується структура); \mathcal{HT} — множин букв, що не збігаються з буквами алфавіту \mathcal{E} і службовців для позначення перемінних; \mathcal{AK} — множин аксіом вирахування, під якими розуміються вихідні формули, що задаються, (слова) в алфавіті \mathcal{E} (наприклад, відповідності функцій і елементів); \mathcal{P} - множин правил висновку нових формул в алфавіті \mathcal{E} з аксіом і раніше виведених коректних формул. Кожну формулу можна інтерпретувати як деяку структуру, тому синтез — це процес висновку формули, що задовольняє вихідним вимогам і обмеженням.

Інші приклади компактного завдання множині альтернатив A через множині \mathcal{E} і \mathcal{P} зв'язаний з використанням систем штучного інтелекту, у яких \mathcal{E} є база даних, \mathcal{P} — база знань, або еволюційних методів, у яких \mathcal{E} — також база даних, \mathcal{P} — множин евристик, послідовність застосування яких визначається еволюційними і генетичними принципами.

23. Методи структурного синтезу в САПР

Системи штучного інтелекту.

У теорії інтелектуальних систем синтез реалізується за допомогою експертних систем (ЭС)

$$\mathcal{ЭС} = \langle \mathcal{БД}, \mathcal{БЗ}, \mathcal{И} \rangle,$$

де $\mathcal{БД}$ — база даних, що включає зведення про базові елементи; $\mathcal{БЗ}$ — база знань, що містить правила конструювання варіантів структури; $\mathcal{И}$ — інтерпретатор, що встановлює послідовність застосування правил із $\mathcal{БЗ}$. Системи штучного інтелекту (СИИ) засновані на знаннях, відділених від процедурної частини програм і представлених в одній з характерних форм. Такими формами можуть бути продукції, фрейми, семантичні мережі. Реально функціонуючі в сучасних САПР системи з базами знань найчастіше відносяться до класу ЭС.

Продукція являє собою правило типу "якщо A , то B ", де A — умова, а B — дію або наслідок, активізує при істинності A . Продукційна $\mathcal{БЗ}$ містить сукупність правил, що описують визначену предметну область.

Фрейм — структура даних, у якій у визначеному порядку представлені зведення про властивості описуваного об'єкта. Типовий вид фрейму:

$$\langle \text{ім'я фрейму}; x_1 = p_1; x_2 = p_2; \dots; x = p; q_1, q_2, \dots, q \rangle,$$

де x_i — ім'я i -го атрибута, p_i — його значення, *засланн- q_i -посилання* на інший фрейм або деяку обслуговуючу процедуру. У якості p_i можна використовувати ім'я іншого (вкладеного) фрейму, описуючи тим самим ієрархічні структури фреймів.

Семантична мережа — форма представлення знань у виді сукупності понять і явно виражених відносин між ними в деякій предметній області. Семантичну мережу зручно представляти у виді графа, у якому вершини відображають поняття, а ребра або дуги — відносини між ними. Як вершини мережі можна використовувати фрейми або продукції.

Експертна система є типовою системою штучного інтелекту, у якій $\mathcal{БЗ}$ містить зведення, отримані від людей-експертів у конкретній предметній області. Труднощі формалізації процедур структурного синтезу

систем

привели до популярності застосування експертних систем у САПР, оскільки в них замість виконання синтезу на базі формальних математичних методів здійснюється синтез на основі досвіду і неформальних рекомендацій, отриманих від експертів.

Дискретне математичне програмування.

Вибір методу пошуку рішення — друга проблема після формалізації задачі. Якщо при формалізації всі керовані параметри удалось представити в числовому виді, то можна спробувати застосувати відомі методи ДМП.

Задача ДМП визначається в такий спосіб:

$$\begin{aligned} & \text{extr } F(\mathbf{X}), \\ & \mathbf{X} \in \mathbf{D} \end{aligned} \quad (4.30)$$

$$\mathbf{D} = \{ \mathbf{X} \mid \mathbf{W}(\mathbf{X}) > 0, \mathbf{Z}(\mathbf{X}) = 0 \},$$

де $F(X)$ — цільова функція; $W(X)$, $Z(X)$ — функції, зв'язані з представленими в ТЗ вимогами й обмеженнями; \mathbf{D} — дискретна множин. В отриманій моделі, по-перше, кожен елемент множині розглянутих закінчених структур повинний мати унікальне сполучення значень деякої множині числових параметрів, вектор яких позначимо \mathbf{X} . По-друге, необхідне існування однієї або декількох функцій $\Phi(\mathbf{X})$, значення яких можуть служити вичерпною оцінкою відповідності структури пропонованим вимогам. По-третє, функції $\Phi(\mathbf{X})$ повинні відбивати внутрішньо властивому даному класові об'єктів властивості, що забезпечить можливість використання $\Phi(\mathbf{X})$ у якості не тільки засобів оцінки досягнутого при пошуку успіху, але і засобів указівки перспективних напрямків продовження пошуку. Ці умови здійсненні далеко не завжди, що й обумовлює труднощі формалізації задач структурного синтезу.

Однак наявність формулювання (4.30) ще не означає, що удасться підібрати метод (алгоритм) рішення задачі (4.30) із прийнятними витратами обчислювальних ресурсів. Іншими словами, застосування точних методів математичного програмування викликає нездоланних труднощів у більшості випадків практичних задач типового розміру через їхню приналежність до класу NP-складних задач. Тому лідируюче положення серед методів рішення задачі (4.30) займають наближені методи, зокрема, декомпозиційні методи, що відбивають принципи блочно-ієрархічного проектування складних об'єктів. Декомпозиційні методи засновані на виділенні ряду ієрархічних рівнів, на кожному з яких вирішуються задачі прийняттого розміру.

Основу великої групи математичних методів, що виражають прагнення до скорочення перебору, складають операції поділу множині варіантів на підмножини і відсікання безперспективних підмножин. Ці методи поєднуються за назвою *методу галузей і границь*. Основний різновид методу галузей і границь відноситься до точних методів рішення комбінаторних задач. Розглянемо цей різновид.

Нехай мається множин рішень \mathbf{M} , у якому потрібно вибрати оптимальний за критерієм $F(\mathbf{X}_j)$ варіант, де \mathbf{X}_j — вектор параметрів варіанта $m_j \in \mathbf{M}$; нехай також мається алгоритм для обчислення нижньої границі $L(\mathbf{M}_k)$ критерію $F(\mathbf{X}_j)$ в будь-якому підмножиною множини \mathbf{M}_k , \mathbf{M} тобто такого значення $L(\mathbf{M}_k)$, що при $F(\mathbf{X}_j) \geq L(\mathbf{M}_k)$ кожному (мається на увазі мінімізація $F(\mathbf{X})$). Тоді основна схема рішення задач відповідно до методу галузей і границь містить наступні процедури: 1) у якості \mathbf{M}_k приймаємо всю множин \mathbf{M} ; 2) розгалуження: розбивка \mathbf{M}_k на кілька підмножин \mathbf{M}_q ; 3) обчислення нижніх границь $L(\mathbf{M}_q)$ у підмножинах \mathbf{M}_q ; 4) вибір у якості \mathbf{M}_k підмножини \mathbf{M}_p із мінімальним значенням нижньої границі критерію (серед усіх підмножин, що маються на даному етапі обчислень), зведення про інші підмножини \mathbf{M}_q і їхніх нижніх границях зберігаються в окремому списку; 5) якщо $|\mathbf{M}_k| > 1$, то перехід до

Лекція 9. Технічне забезпечення процесу проектування процедури 2, інакше одноелементна множин \mathbf{M}_k є рішення.

Метод галузей і границь у випадку точного обчислення нижніх границь відноситься до точних методів рішення задач вибору і тому в несприятливих ситуаціях може приводити до експонентної тимчасової складності. Однак метод часто використовують як наближений, оскільки можна застосовувати наближені алгоритми обчислення нижніх границь.

Серед інших наближених методів рішення задачі ДМП відзначимо *метод локальної оптимізації*. Тому що простір D метризовано, те можна використовувати поняття a -околиці $S_a(X_k)$ поточної крапки пошуку X_k . Замість перебору крапок у всьому просторі D здійснюється перебір крапок тільки в $S_a(X_k)$. Якщо $F(X_j) \geq F(X_k)$ для всіх $X_j \in S_a(X_k)$, те вважається, що знайдено локальний мінімум цільової функції в крапці X_k . У протилежному випадку крапку X_q , у якій досягається мінімум $F(X)$ у $S_a(X_k)$, приймають у якості нової поточної крапки пошуку.

Елементи теорії складності.

У теорії складності виділяють масові й індивідуальні задачі. Перші з них сформульовані в загальному виді, другі представлені з конкретними числовими значеннями вихідних даних. Дослідження складності проводяться у відношенні масових задач і одержувані висновки, як правило, відносяться до найгіршого випадку — до найбільш несприятливого можливого сполучення вихідних даних.

Ціль досліджень — установлення виду залежності обсягу Q необхідних обчислень від розміру задачі N . Обсяг обчислень може визначатися числом арифметичних і логічних операцій або витратами процесорного часу ЕОМ із заданою продуктивністю. Розмір задачі в загальному випадку зв'язують з обсягом опису задачі, але в додатках поняття розміру легко наповняється більш конкретним змістом.

Далі, у теорії складності задач вибору вводять поняття ефективних і неефективних алгоритмів. До *ефективного* відносять алгоритми з поліноміальною залежністю Q від N , наприклад, алгоритми з функцією $Q(N)$ лінійної, квадратичної, кубічної й ін. Для *неефективних* алгоритмів характерна експонентна залежність $Q(N)$.

Важливість проведення різкої границі між поліноміальними й експонентними алгоритмами впливає з зіставлення числових прикладів росту припустимого розміру задачі зі збільшенням швидкодії B використовуваних ЕОМ (табл. 4.1, у якій зазначені розміри задач, розв'язуваних за те саме час T на ЕОМ зі швидкодією B_1 при різних залежностях складності Q від розміру N). Ці приклади показують, що вибираючи ЕОМ у *До* раз більш швидкодіючу, одержуємо збільшення розміру розв'язуваних задач при лінійних алгоритмах у *До* раз, при квадратичних алгоритмах у $K^{1/2}$ разів і т.д.

Таблиця 4.1

$Q(N)$	B_1	$B_2=100B_1$	$B_3=1000B_1$
N	N_1	$100N_1$	$1000N_1$
N^2	N_2	$10N_2$	$31.6N_2$
N^3	N_3	$4.64N_3$	$10N_3$
2^N	N_4	$6.64+N_4$	$9.97+N_4$

Інакше обстоїть справа з неефективними алгоритмами. Так, у випадку складності 2^N для того самого процесорного часу розмір задачі збільшується тільки на $\lg \lg 2$ одиниць. Отже, переходячи від ЕОМ з $B = 1$ Gflops до суперЕВМ із $B = 1$ Tflops, можна збільшити розмір розв'язуваної задачі тільки на 10, що зовсім недостатньо для практичних задач. Дійсно, у таких задачах, як наприклад, синтез тестів для БІС число вхідних двоичних перемінних може складати більш 150 і тому повний перебір усіх можливих кодів, що перевіряють, зажадає виконання більш 2150 варіантів моделювання схеми.

У теорії складності всі комбінаторні задачі розділені на класи:

- клас нерозв'язних задач, у котрій входять масові задачі, рішення яких повним перебором принципово неможливо з погляду сучасних наукових представлень; цей клас відокремлюється від інших задач так названою межею Бреммермана, оцінюваном величиною $N = 10^{93}$, відзначимо, що реальна межа нерозв'язності значно нижче;
- клас P, до якого відносяться задачі, для яких відомі алгоритми рішення поліноміальної складності;
- клас NP, включаючий задачі, для яких можна за поліноміальний час перевірити правильність рішення, тобто відповісти на запитання, чи задовольняє дане рішення заданим умовам; очевидно, що P включено в NP, однак питання про збіг цих класів поки залишається відкритим, хоча очевидно на це питання буде отримана негативна відповідь;
- клас NP-повних задач, що характеризується наступними властивостями: 1) для цих задач не відомі поліноміальні алгоритми точного рішення; 2) будь-які задачі усередині цього класу можуть бути зведені одна до іншої за поліноміальний час. Останнє означає, що якщо буде знайдений поліноміальний алгоритм для точного рішення хоча б однієї NP-повної задачі, то за поліноміальний час можна буде вирішити будь-яку задачу цього класу.

З результатів теорії складності випливають важливі практичні рекомендації: 1) приступаючи до рішення деякої комбінаторної задачі, варто спочатку перевірити, чи не належить вона до

класові NP-повних задач, і якщо це так, то не слід витратити зусилля на розробку алгоритмів і програм точного рішення; 2) відсутність ефективних алгоритмів точного рішення масової задачі вибору аж ніяк не означає неможливості ефективного рішення індивідуальних задач із класу NP-повних або неможливості одержання наближеного рішення по евристичних алгоритмах за поліноміальний час.

Еволюційні методи.

Еволюційні методи (ЕМ) призначені для пошуку кращих рішень і засновані на статистичному підході до дослідження ситуацій і ітераційному наближенні до шуканого стану систем.

На відміну від точних методів математичного програмування ЕМ дозволяють знаходити рішення, близькі до оптимальних, за прийнятний час, а на відміну від відомих евристичних методів оптимізації характеризуються істотно меншою залежністю від особливостей додатка (тобто більш універсальні) і в більшості випадків забезпечують кращий ступінь наближення до оптимального рішення. Універсальність ЕМ визначається також застосовністю до задач з неметризуємим простором керованих перемінних (тобто серед керованих перемінних можуть бути і лінгвістичні).

Найважливішою часткою случаємо ЕМ є *генетичні методи й алгоритми*. Генетичні алгоритми (ГА) засновані на пошуку кращих рішень за допомогою спадкування і посилення корисних властивостей множині об'єктів визначеного додатка в процесі імітації їхньої еволюції.

Властивості об'єктів представлені значеннями параметрів, поєднуваними в запис, називаний у ЕМ *хромосома*. У ГА оперують хромосомами, що відносяться до множині об'єктів — *популяції*. Імітація генетичних принципів — вероятностний вибір батьків серед членів популяції, схрещування їхніх хромосом, добір нащадків для включення в нові покоління об'єктів на основі оцінки цільової функції — веде до еволюційного поліпшення значень цільової функції (функції корисності) від покоління до покоління.

Серед ЕМ знаходять застосування також методи, що на відміну від ГА оперують не множинцю хромосом, а єдиною хромосомаю. Так, метод дискретного *локального пошуку* (його англійська назва *Hillclimbing*) заснований на випадковій зміні окремих параметрів (тобто значень полів у записі або, іншими

Лекція 9. Технічне забезпечення процесу проектування
 словами, значень генів у хромосомі). Такі зміни називають *мутаціями*. Після чергової мутації оцінюють значення *функції корисності* F (Fitness Function) і результат мутації зберігається в хромосомі тільки, якщо F покращилася. В іншому ЕМ за назвою "*Моделювання отжигу*" (Simulated Annealing) результат мутації зберігається з деякою імовірністю, що залежить від отриманого значення F .

Постановка задачі пошуку оптимальних рішень за допомогою генетичних алгоритмів.

Для застосування ГА необхідно:

- 1) виділити сукупність властивостей об'єкта, що характеризуються внутрішніми параметрами і впливаючих на його корисність, тобто виділити множин керованих параметрів $X = (x_1, x_2, \dots, x_n)$; серед x_i можуть бути величини різних типів (real, integer, Boolean, enumeration). Наявність нечислових величин (enumeration) обумовлює можливість рішення задач не тільки параметричної, але і структурної оптимізації;
- 2) сформулювати кількісну оцінку корисності варіантів об'єкта — функцію корисності F . Якщо у вихідному виді задача многокритеріальна, то таке формулювання означає вибір скалярного (узагальненого) критерію;
- 3) Розробити математичну модель об'єкта, що представляє собою алгоритм обчислення F для заданого вектора X ;

Представити вектор X у формі хромосоми — запису наступного виду

X_1	X_2	X_3	...	X_n
-------	-------	-------	-----	-------

У ГА використовується наступна термінологія:

ген — керований параметр x_i ;

аллель — значення гена;

локус (позиція) — позиція, займана геном у хромосомі;

генотип — екземпляр хромосоми, генотип представляє сукупність внутрішніх параметрів проєктованого за допомогою ГА об'єкта;

генофонд — множин усіх можливих генотипів;

функція корисності (пристосованості) F — цільова функція;

фенотип — сукупність генотипу і відповідного значення F , під фенотипом часто розуміють сукупність вихідних параметрів синтезованого за допомогою ГА об'єкта.

Простий генетичний алгоритм.

Обчислювальний процес починається з генерації вихідного покоління — множині, що включає N хромосом, N — розмір популяції. Генерація виконується випадковим вибором аллелей кожного гена.

Далі організується циклічний процес зміни поколінь:

```
for (k=0; k<G; k++) { for (j=0; j<N; j++)
{ Вибір батьківської пари хромосом;
Кроссовер;
Мутації;
Оцінка функції корисності F нащадків;
Селекція;
}
Заміна поточного покоління новим;
```

систем

}

Для кожного витка зовнішнього циклу генетичного алгоритму виконується внутрішній цикл, на якому формуються екземпляри нового (наступного за поточним) покоління. В внутрішньому циклі повторюються оператори вибору батьків, кроссовера батьківських хромосом, мутації, оцінки пристосованості нащадків, селекції хромосом для включення в чергове покоління.

Розглянемо алгоритми виконання операторів у простому генетичному алгоритмі.

Вибір батьків. Цей оператор імітує природний добір, якщо добір у батьківську пару хромосом із кращими значеннями функції корисності F більш ймовірний. Наприклад, нехай F потрібно мінімізувати. Тоді ймовірність P_i вибору батька з хромосою C_i можна розрахувати по формулі

$$P_i = (F_{\max} - F_i) / \sum_{j=1}^N (F_{\max} - F_j) \quad (4.31)$$

де F_{\max} — найгірше значення цільової функції F серед екземплярів (членів) поточного покоління, F_i — значення цільової функції i -го екземпляра.

Правило (4.31) називають *правилом колеса рулетки*. Якщо в колесі рулетки виділити сектори, пропорційні значенням $F_{\max} - F_i$, то ймовірності влучення в них суть P_i , обумовлені відповідно до (4.31).

Приклад. Нехай $N=4$, значення F_i і P_i приведені в табл. 4.2.

Таблиця 4.2

i	F_i	$F_{\max} - F_i$	P_i
1	2	5	0,5
2	1	0	0
3	6	1	од
4	3	4	0,4

Кроссовер (схрещування). Кроссовер, іноді називаний кроссинговером, полягає в передачі ділянок генів від батьків до нащадків. При простому (однокрапковому) кроссовері хромосоми батьків розриваються в деякій позиції, однакової для обох батьків, вибір місця розриву рівновероятен, далі відбувається рекомбінація частин батьківських хромосом, що утворюються, як це показано в табл. 4.3, де розрив мається на увазі між п'ятим і шостим локусами.

Таблиця 4.3

Хромосома	Гени							
батька А	f	a	c	d	g	k	V	e
батька В	a	b	c	d	e	/	g	h
нащадка З	f	a	c	d	g	/	g	h

Лекція 9. Технічне забезпечення процесу проектування

нащадка D	a	b	c	d	e	k	V	e
-----------	---	---	---	---	---	---	---	---

Мутації. Оператор мутації виконується з деякою імовірністю P_m , тобто з імовірністю P_m відбувається заміна аллеля випадковим значенням, обраним з рівною імовірністю в області визначення гена. Саме завдяки мутаціям розширюється область генетичного пошуку.

Селекція. Після кожного акта генерації пари нащадків у нове покоління включається кращий екземпляр пари.

Внутрішній цикл закінчується, коли число екземплярів нового покоління стане рівним N . Кількість повторень G зовнішнього циклу найчастіше визначається автоматично по появі ознак виродження (стагнації) популяції, але з умовою не перевищення заданого ліміту машинного часу.

Різновиди генетичних операторів.

Можливі відхилення від представленої вище в простому генетичному алгоритмі схеми обчислень.

Кроссовер. По-перше, припустимі схеми багатоточкового кроссовера.

По-друге, відзначимо ситуації, коли на склад аллелей накладені деякі додаткові умови. Наприклад, нехай у задачі розбивки графа число вершин у підграфах A_1 і A_2 повинно бути N_1 і N_2 і нехай k -й аллель, рівний 1, означає, що вершина k попадає в A_1 , якщо ж k -й аллель дорівнює ПРО, то в A_2 . Очевидно, що число одиниць у хромосомі повинне дорівнювати N_1 , число нулів — N_2 . Тоді при рекомбінації ліва ділянка хромосоми береться від одного з батьків без змін, а в правій ділянці (від іншого батька) потрібно погодити число одиниць з N_1 тим або іншому способом.

Один зі способів — метод РМХ (Partially Matched Crossover). Для ілюстрації РМХ розглянемо приклад двухточкового кроссовера в задачі, коли в хромосомі повинні бути присутнім, причому тільки по одному разі, усі значення генів із заданого набору. Нехай у прикладі цей набір включає числа від 1 до 9.

Таблиця 4.4.

1	2	3	4	5	6	7	8	9
3	7	1	9	2	4	8	6	5
1	2	1	9	2	6	7	8	9
1	2	3	9	5	6	7	8	4

У табл. 4.4 перші двох рядків складають батьківські хромосоми. Третій рядок містить хромосому одного з нащадків, згенерованого в результаті застосування двухточкового кроссовера (після другого і п'ятого локусів). Отримана хромосома не відноситься до числа припустимих, тому що в ній значення генів 1, 2 і 9 зустрічаються двічі, а значення 3, 4 і 5 відсутні. Четвертий рядок показує результат застосування РМХ. У цьому методі виділяються сполучені пари аллелей в однойменних локусах однієї з рекомбинируємих частин. У нашому прикладі це пари (3 і 1), (4 і 9), (5 і 2). Хромосома нащадка проглядається ліворуч праворуч; якщо повторно зустрічається деяке значення, воно замінюється на сполучене значення. Так, у прикладі в локусах 3, 5 і 9 повторно зустрічаючі аллели 1, 2 і 9 послідовно замінюються на значення 3, 5 і 4.

Мутації. Бувають крапковими (в одному гені), макромутаціями (у декількох генах) і хромосомними (поява нової хромосоми). Звичайно імовірність появи мутації вказується серед вихідних даних. Але можливо автоматичне регулювання числа мутацій при їхній реалізації тільки в ситуаціях, коли батьківські

систем

хромосоми розрізняються не більш ніж у D_0 генах.

Селекція. Після визначення і позитивної оцінки нащадка він може бути відразу ж включений у поточну популяцію замість гіршого зі своїх батьків, при цьому з алгоритму виключається зовнішній цикл (що однак не означає скорочення загального обсягу обчислень).

Інший варіант селекції — добір після кожної операції схрещування двох кращих екземплярів серед двох нащадків і двох батьків.

Часто член популяції з мінімальним (кращим) значенням цільової функції примусово включається в нове покоління, що гарантує спадкування придбаних цим членом позитивних властивостей. Такий підхід називають *елітизмом*. Звичайно елітизм сприяє більш швидкої збіжності до локального екстремуму, однак у многоекстремальної ситуації обмежує можливості влучення в околиці інших локальних екстремумів.

Примітка. Хромосому X^* будемо називати точкою локального мінімуму, якщо $F(X^*) < F(X_i)$ для всіх хромосом X_i , що відрізняються від X^* значенням єдиного гена, де $F(X)$ — значення функції корисності в крапці X .

Наступний варіант селекції — добір N екземплярів серед членів репродукційної групи, що складається з батьків, нащадків і мутантів, що задовольняють умові $F_i < t$, де t — граничне значення функції корисності. Поріг може бути дорівнює або середньому значенню F у поточному поколінні, або значенню F особи, що займає визначене порядкове місце. При цьому м'яка схема добору — у нове покоління включаються N кращих представників репродукційної групи. Тверда схема добору — у нове покоління екземпляри включаються з імовірністю q_i :

$$q_i = (F_{\max} - F_i) / \sum_{j=1}^{N_r} (F_{\max} - F_j)$$

де N_r — розмір репродукційної групи.

Переупорядкування. Крім перерахованих основних операторів, знаходять застосування деякі додаткові. До їхнього числа відноситься оператор переупорядкування генів — зміни їхнього розподілу по локусах.

Призначення переупорядкування зв'язане з властивістю, що носить назву епістасис. *Епістасис* має місце, якщо функція корисності залежить не тільки від значень генів (аллелей), але і від їхній позиціонування. Наявність епістасиса говорить про нелінійність цільової функції й істотно ускладнює рішення задач. Дійсно, якщо деякі аллелі двох генів роблять визначений позитивний вплив на цільову функцію, утворити деяке зв'язування (схему), але внаслідок епістасиса при розриві зв'язування ці аллелі роблять уже протилежний вплив на функцію корисності, то розривати такі схеми не впливає. А це означає, що зв'язані епістасисом гени бажано розташовувати близько друг до друга, тобто при невеликих довжинах схем. Оператор переупорядкування допомагає автоматично "намацати" такі сукупності генів (вони називаються хромосомними блоками або building blocks) і розмістити них у близьких локусах.

Генетичний метод комбінування евристик.

Можливі два підходи до формування хромосом.

Перший з них заснований на використанні як гени проектних параметрів. Наприклад, у задачі розміщення мікросхем на платі локуси відповідають посадковим місцям на платі, а генами є номери (імена) мікросхем. Іншими словами, значенням k -го гена буде номер мікросхеми в k -й позиції.

В другому підході генами є не самі проектні параметри, а номери евристик, використовуваних для визначення проектних параметрів. Так, для задачі розміщення можна застосовувати кілька евристик. По одній з них у чергове посадкове місце потрібно поміщати мікросхему, що має найбільше число зв'язків із уже

Лекція 9. Технічне забезпечення процесу проектування

розміщеними мікросхемами, по іншій — мікросхему з мінімальним числом зв'язків із ще не розміщеними мікросхемами і т.д. Генетичний пошук у цьому випадку є пошук послідовності евристик, що забезпечує оптимальний варіант розміщення.

Другий підхід одержав назву — *метод комбінування евристик*. Цей метод виявляється кращим у багатьох випадках. Наприклад, у задачах синтезу розкладів розподіляється задана множина робіт у часі і між обслуговуючими пристроями — серверами, тобто проектними параметрами для кожної роботи будуть номер сервера і порядковий номер у черзі на обслуговування. Нехай N — число робіт, M — число серверів. Якщо гени відповідають номерам робіт, то в першому підході в хромосомі потрібно мати $2N$ генів і загальне число отличаючихся друг від друга хромосом W помітно перевищує найбільше з чисел $N!$ і M^N .

Відповідно до методу комбінування евристик, число генів у хромосомі в два рази менше, ніж у першому підході, і дорівнює N . Тому якщо число використовуваних евристик дорівнює DO , то потужність множині можливих хромосом уже незрівнянно менше, а саме

$$W = KN.$$

Очевидно, що менший розмір хромосоми веде до кращої обчислювальної ефективності, а менше значення W дозволяє швидше знайти околиці шуканого екстремуму. Крім того, у методі комбінування евристик усі хромосоми, генерируемые при кроссовере, будуть припустимими. У той же час при застосуванні звичайних генетичних методів необхідно використовувати процедури типу РМХ для коректування генів, що відносяться до номерів у черзі на обслуговування, що також знижує ефективність пошуку.

Частина IV.

Автоматизоване проектування МСТ в IntelliSuite

24. Структура IntelliSuite

Overview



Living Design Environment

IntelliSuite is a tightly integrated design environment that will link your entire MEMS organization together. Built to scale from a point tool to an organization-wide tool, IntelliSuite unifies various engineering and manufacturing tasks into a single living design environment.

Designed around collaboration, IntelliSuite allows the design, process, packaging and system teams collaborate on MEMS devices that can be prototyped and manufactured with fewer costly iterations.

Industry standard

Synple Schematic capture Component based Design exploration Mask & 3D mesh synthesis	Blueprint Physical design Layout/DRC Cross sectioning Output to 3D FEA	Clean Room Process flow design Process debug Process visualization Output to 3D FEA	Fast Field Multiphysics solvers Coupled field analysis System model extraction	EDA Linker Link to EDA tools Cadence, Mentor, Synopsys, Ansoft, Mathworks etc...

After close to two decades in continuous development, IntelliSuite now boasts the widest range of features available in a MEMS tool, and remains unsurpassed in its simulation capabilities, robustness and ease of use. That's why IntelliSuite serves users in more than 35 countries, including majority of the top MEMS manufacturers, research organizations and MEMS universities. And that's why it's the industry standard software for MEMS.

246. Лекційний курс: «Основи автоматизованого проектування складних об'єктів і систем»

Seamless design flow

IntelliSuite contains a wide range of closely integrated tools to seamlessly go from schematic capture and optimization to design verification and tapeout. A flexible design flow allows you to start your design at either schematic, layout or 3D level.

IntelliSuite consists of a number of advanced tools that work together. For instance, Synple allows you to capture your MEMS at a schematic level, much like SPICE for electrical circuits. Your design can then be quickly iterated and optimized at different granularities. Sophisticated synthesis algorithms can automatically convert your schematic into mask layout, 3D or better yet a meshed structure for full multiphysics analysis.

Blueprint, is a physical design tool that incorporates advanced layout, design rule check, cross section exploration, and automated mask to hex mesh capabilities. CleanRoom process suite allows you to create and debug your process flow and your mask set before you enter the clean room. It allows you to make virtual prototypes to save costly fabrication mistakes. Advanced Fastfield Multiphysics tools feature fully coupled electrostatic, mechanical, fluidic, and electromagnetic engines. Advanced Model Order Reduction based Extraction techniques capture electromechanical, fluidic and damping behavior into compact models, while EDA Linker seamlessly links IntelliSuite to other leading EDA tools.

Design the next small thing ...



IntelliSuite comes with all the design tools required to take your product from concept to market. We ship a full complement of tools with which you can perform process modeling, device layout and device analysis. In fact, IntelliSuite comes standard with completely coupled electrical, piezo-electrical, mechanical and thermal analysis tools for executing linear or non-linear static or transient analysis, while optional modules for electromagnetic, fluidic and system analysis are also available.

At IntelliSense, we constantly monitor the latest developments in the MEMS industry for new features and capabilities. And if we haven't already provided everything you're looking for, we can develop customized features to meet your specific needs.

N/MEMS Design Methodologies

N/MEMS Design Today

As the N/MEMS industry matures, the design challenge continues to move from the microstructure design to the microsystem design. With the maturity of the process technology and the increase in computing power, designers are now looking to optimize MEMS from a system standpoint. Traditional N/MEMS CAD tools provide functionality to design at a microstructure level.

Today MEMS modeling and simulation is performed at various levels of granularity by MEMS engineers working on different aspects of the manufacture. Ab initio models are based upon atomistic, quantum mechanical or molecular dynamics. Such models are typically used in process

систем

modeling to predict material behavior (such as physical properties or etch behavior). Component level models can include lumped models and finite element representations of a component such as a plate or a comb drive. Device models represent the working of the micro or nanostructure under investigation. Algorithmic models are used to capture the behavior of a certain logic or control element within a system. Finally, system level models are used to model the entire microsystem.

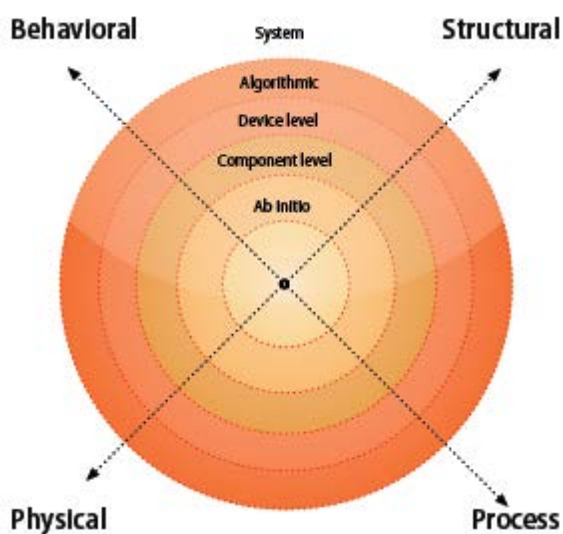


Рис. 1. X-Diagram for Micro and Nanosystem development, illustrating various levels of granularity of the design.

Ab initio (first principles) simulations

First principle simulations are typically based on atomistic or quantum mechanical molecular dynamics principles. These are typically useful in predicting material properties or predicting phenomenon at the nanoscale. While, this is a growing area of research, only a few tools have made it into the everyday repertoire of the MEMS design engineer. One such tool is IntelliSense's IntelliEtch which uses atomistic principles to simulate the etching of silicon.

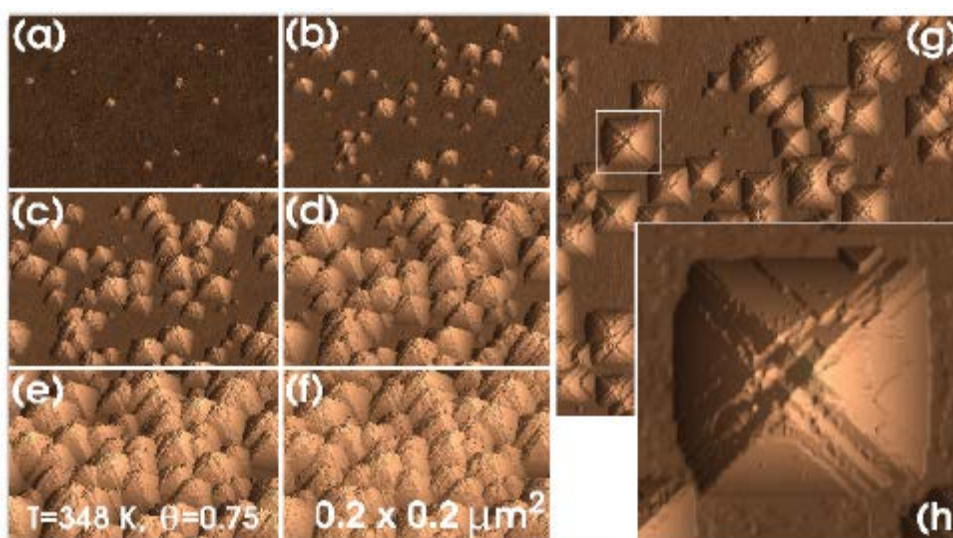


Рис. 2. Atomistic calculations are used to predict hillock formation and surface morphology during wet etching of silicon.

Лекція 9. Технічне забезпечення процесу проектування

Ab initio techniques allow the user to capture effects of micro-masking which can lead to hillock formation, preventing smooth etches.

Technology CAD (TCAD)

At this level, the microstructure is simulated at the process level. Simulators such as AnisE™ and RECIPE™ from IntelliSense and SUPREM™-based simulators from various vendors simulate the actual process flow based upon process settings and physical simulation of the process, such as diffusion, growth or etching. TCAD-based models are typically set up and run by process engineers.

These simulations are useful in understanding the effect of the process on the final physical geometry of the device. Since they are based on the actual physical models, they are often very time-consuming. For instance, IntelliSense's RIE/ICP simulation tool RECIPE is based on the actual simulation of the plasma etching process and polymer deposition process. These tools are used to determine the influence of the process and mask set on the final geometry of the device.

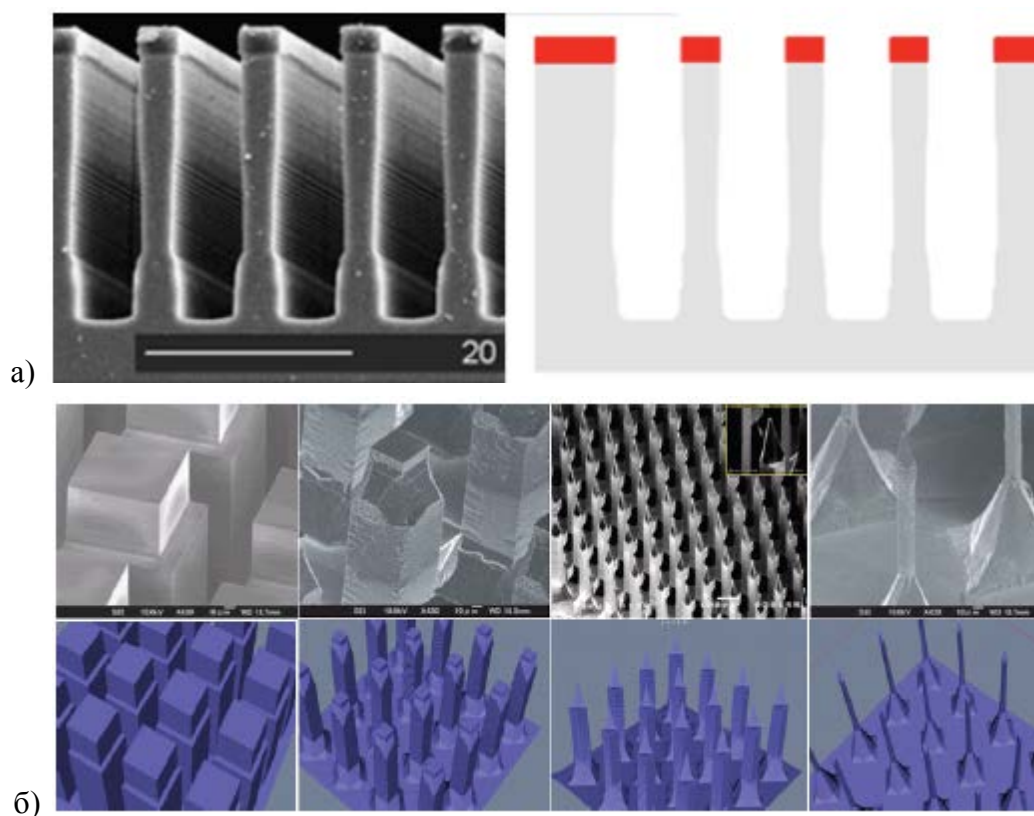


Рис. __. Technology CAD (TCAD) tools are used to accurately predict the physical etching and processing of MEMS devices: a) comparison between experiment and simulation of Deep Reactive Ion Etching (DRIE) of silicon. Sidewall angle tuning is integral for high performance of many classes of MEMS devices; б) A complex process of formation of micro-needles being simulated in software.

Process flow design (Process CAD)

The bane of every MEMS process engineer is creating and recreating flow diagrams (2D cross sections) of the process flow, while some process engineers still use 2D drawing tools such as Adobe Illustrator or PowerPoint. Increasingly sophisticated tools such as FABViewer and CSVviewer (integrated into IntelliSense's Blueprint) have started to address this market. These tools can be used to output process flows directly into Powerpoint or other presentation tools.

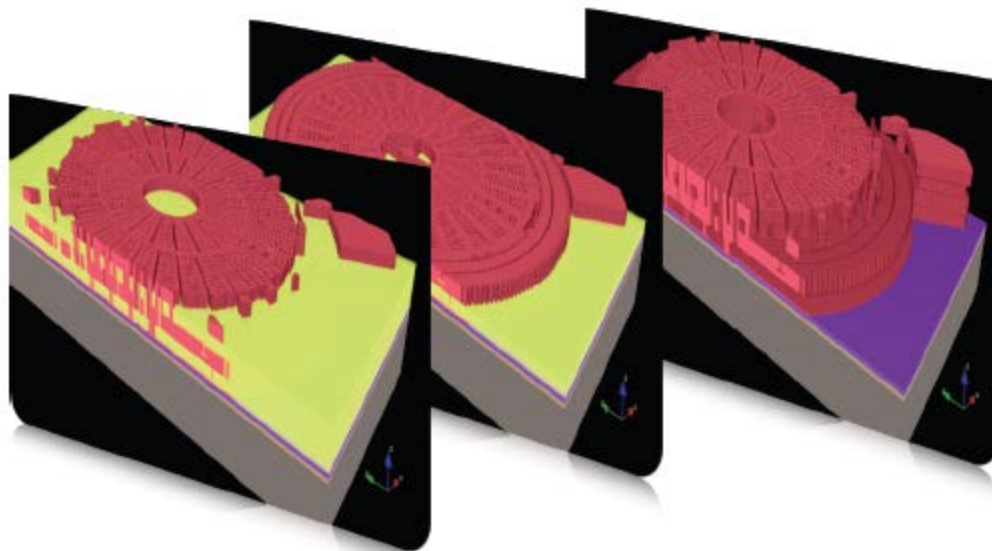


Рис. __. Snapshots from the process visualization of a rotary Vibro-drive fabricated using the Sandia Summit-V process. Courtesy: Prof Tim Dallas

Schematic or component based design (Top-down design)

One of the primary advantages behind a hierarchical approach is that the design entry is done in terms of fundamental building blocks or components. This allows the user to enter a parameterized model of the device in terms of both layout and manufacturing data.

Since the data entry is done in terms of parameterized abstract models, users can analyze the devices at different granularities. The atomic element model can be represented in terms of lumped models, distributed models, or Rayleigh-Ritz based FEA/BEA models. The user can then easily perform an accuracy-time trade-off.

One of the disadvantages of schematic-based design is that the user is limited to using components in the design library. Arbitrary geometries and new physical or material models are difficult to incorporate into the design. Since most schematic models are, to a degree, based on lumped models, they cannot accurately capture second-order effects and non-linearities.

For instance, accurate capture of electrostatics, fluidics or contact and post-contact physics in arbitrary geometries requires full 3D modeling. Similarly, packaging level effects such as influence of viscoelastic overmolds, effects of die bumping and substrate attach are difficult to capture in component-based models.

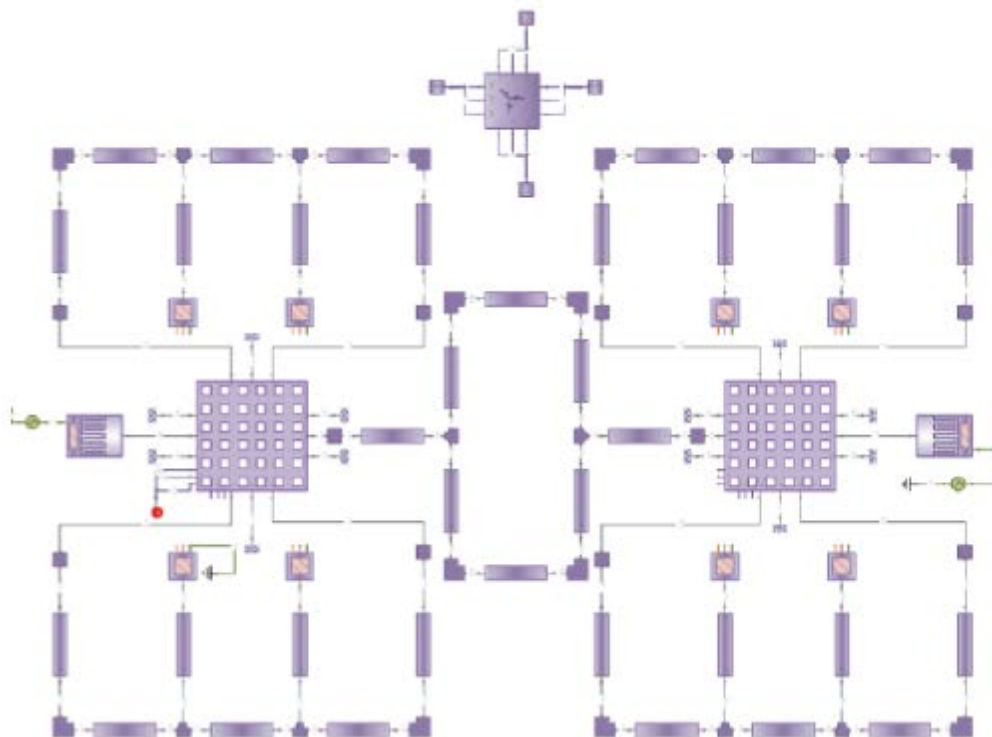


Рис. __. Schematic of a bandpass filter in SYNPLE. SYNPLE allows you to quickly setup a parametric model of your MEMS device.

The figure below shows the results of a Monte Carlo simulation plotting the anticipated variation of natural frequency. Use of compact models allows users to develop robust designs that are inherently manufacturable.

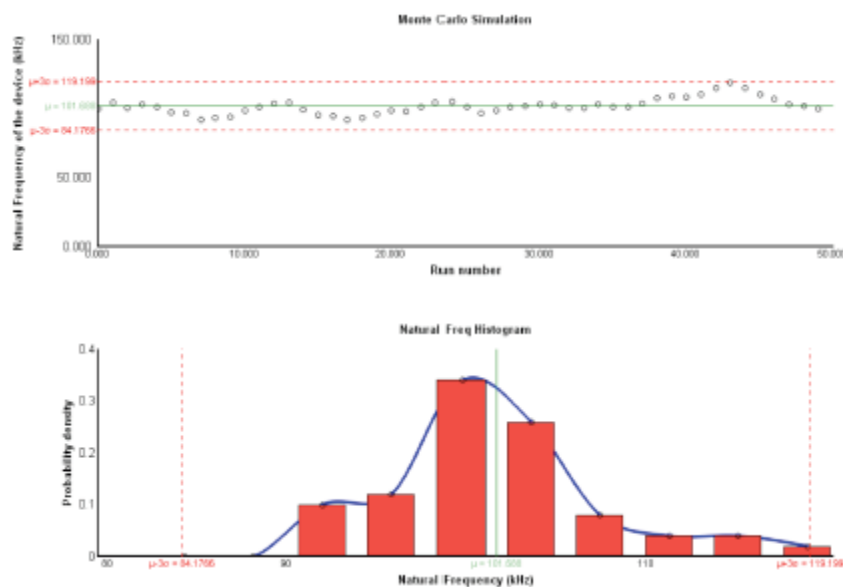


Рис. __.

Schematic synthesis

Schematic synthesis tools, such as those incorporated in SYNPLE can convert component-based schematics into ready-to-use mask layouts, or hexahedral meshes that can be further used in 3D analysis of microsystems.

Layout-based design (Bottom-up design)

3D design, typically the entry mode for the mechanical designer, is still the most popular methodology for MEMS design. Originally pioneered by IntelliSense in the early 90's, this still remains the most popular methodology for MEMS design today.

Layout-based design combines the 2D mask layout with the process flow to create 3D solid models of the MEMS device. These solid models are discretized and analyzed using 3D Finite Element/Boundary Element Analysis (FEA/BEA) methods. Layout-based design has been tremendously successful in microstructure design because it combines design intent with manufacturing.

In addition, bottom-up design can fully capture the complex multiphysics inherent in MEMS devices.

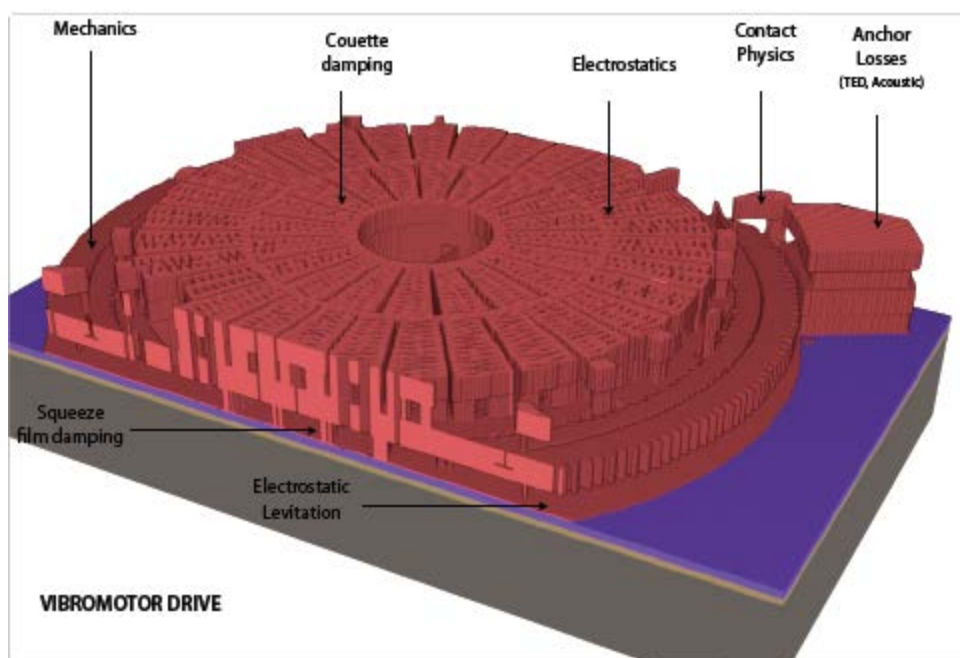


Рис. __. MEMS design is inherently multidomain in nature. The figure depicts the different kind of models that need to be incorporated in the modeling of a vibro-drive. FEA/BEA tools are typically used to capture the multi-physical effects in MEMS devices.

System model extraction (SME)

The layout-based discretized 3D models need to be converted into system models. Typical MEMS 3D models can contain between 100,000 – 1,000,000 degrees of freedom, while system simulators are not designed to handle such complex problems.

Many designers use lumped model approximations to represent the microstructure in system simulations. While these are sufficient for proof-of-concept analysis, they grossly simplify real-world effects such as etching effects, stresses in beams and suspensions, and levitation effects due to charge reflectance. For instance, sidewall angle due to etching can lead to large quadrature errors in inertial devices, residual stresses in beams can cause stiffening of beams, and levitation effects in comb drives can lead to lowered sensitivity of devices.

Лекція 9. Технічне забезпечення процесу проектування

A new class of numerical algorithms based on Krylov/Arnoldi subspace reduction techniques have been developed in the recent years to convert FEA models into arbitrary degree of freedom (NDOF) models. These algorithms are used to capture the total energy and energy dissipation in the system. Based on this, FEA/BEA models can be reduced to efficient compact system models that can be incorporated into system simulators. These compact system models, also known as Reduced Order Models (ROMs), can simulate the microdevice response to within 2-5% of the FEA/BEA model at a 100-1000X performance gain.

The advantage of using ROMs is that they accurately capture the device behavior across multiple energy domains (mechanical, thermal, electrostatics, fluidics, damping, etc). The ROMs can be typically exported into a variety of Hardware Description Languages (HDLs) for use with electrical or system simulators.

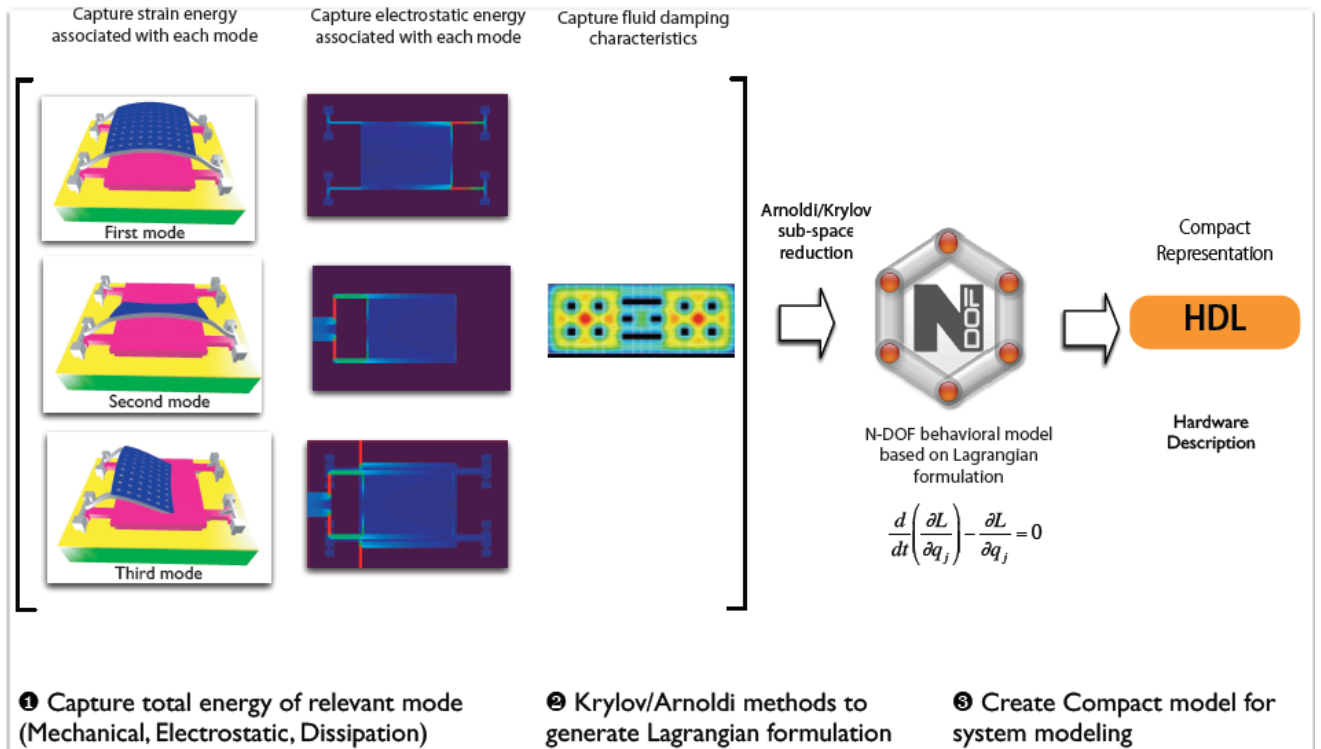


Рис. __. Methodology for efficient extraction of compact models. IntelliSuite creates Look Up Table based Lagrangian ROMs by capturing the energy in each of the physical domains (i.e. mechanical, electrostatic, fluidic, etc).

Verification

Verification in MEMS is quite different from that in the IC world. The IC world typically uses Layout vs. Schematic (LVS) and Design Rule Check (DRC) techniques. While Design Rule Checks can be used in the MEMS world, support for curves, beziers, and all-angle geometries are needed. LVS provides little benefit to the MEMS designer due to the inherent 3D nature of the design.

Schematic vs 3D comparisons (SV3D) are needed to make sure that the schematic capture has been accurately translated into a 3D design. In MEMS design, verification is the process of comparing the results from the schematic-based model (top-down approach) with the results of the 3D-based approach (bottom-up approach). This typically involves benchmarking the schematic, 3D finite element and ROM results.

System simulation

Accurate system simulation can be performed using ROMs described in the previous section. However, compact models are not easily parameterized. While the process of creating a large number of compact models can be automated, it is time-consuming.

Another alternative is to use lumped parameter models. However, these models do not take into account process-related effects such as axial residual stresses, strain gradients, temperature coefficient issues, physical issues such as joule heating, squeeze film and Couette damping or package-level effects of die attach, plastic over-molding and non-uniform heating.

The designer is often faced with the choice of using low-fidelity lumped models which can be parameterized, or high-fidelity nonparameterized compact models. This makes the design optimization from a system standpoint a challenge.

Bottlenecks in current design flows

We can classify bottlenecks in MEMS design according the different stages of the design evolution. The bottlenecks in current MEMS design methodologies are summarized below.

Schematic and System level

- Process information: Ability to easily incorporate process tolerances and process corners. SPC information is often not available for Monte Carlo simulations.

- High-fidelity models: Ability to include high-fidelity models which fully capture multiphysics effects. Users need to capture effects of packaging, fluidic damping, temperature gradients and other non-idealities such as electrostatic gradients or stiction that can greatly influence the device performance.

- Limited libraries: Typical design libraries are limited in nature, and adding new components is a significant undertaking. While complex structures can be composed using basic shapes, schematic layout becomes an issue. Many MEMS schematic editors are little more than warmed-over IC design tools and are inefficient for MEMS design.

- Ability to work with your favorite toolset: Lack of true standards among EDA vendors has led to the creation of multiple quasi-standards, each of which has significant disadvantages. MEMS tools must support many of these quasi-standards including SPICE and its variants (PSPICE, HSPICE, ELDO), Verilog (-A, -AMS, SystemVerilog), VHDLAMS (accounting for subtle but significant differences between tools) , and Matlab/Simulink.

- Co-simulation with other system simulators: While system-level simulators such as SYNPLE or SystemVision provide sophisticated functionality, there is a need for cosimulation along with other system simulators which may contain IP blocks or functionality not present in the tool of choice. The ability to mix system simulators such as SYNPLE with Simulink or Spectre will be extremely useful.

Physical level

- Layout tools: Popular layout tools like Virtuoso or L-Edit were originally developed for the IC world. Other tools such as AutoCAD were developed for architectural and mechanical drawing and are a bear to use for mask layout. Beziers, splines, and smooth transitions (essential for minimizing

Лекція 9. Технічне забезпечення процесу проектування

stresses) are often absent or an afterthought in MEMS layout editors.

- High quality layout automation: While schematic to mask synthesis exists in various tools, the resultant layouts are not production ready. Features such as smooth transitions, stress relief structures, release hole patterns, dimples and other secondary features need to be manually added to the layouts. This is often very timeconsuming.

- Geometry manipulation and meshing: Design analysts spend a significant amount of time manipulating design geometries and meshing structures. Most MEMS design tools require manually partitioning 3D geometry into mesh-able regions. While many tools provide automated tetrahedral meshes, they are sub-optimal for MEMS design. In the words of a Sandia Labs design analyst, “The only people who use tets are those who don’t care about their answers”.

Hexahedral meshes are ideally used for MEMS. The automatic generation of hex meshes has been the holy grail of the meshing community and is still a few years away.

- Parametric meshing: Any parametric changes to the MEMS geometry such as changing a film thickness or electrode gap requires a complete re-meshing of the structure. Parametric meshing and mesh morphing technologies are relatively new and are just making their way into MEMS design tools.

- Incorporation of process corners into physical design: As mentioned before, absence of reliable SPC data forces the designer to consider process corners based upon historical estimates of process tolerances. Users must manually configure their meshes for each of the process corners and run each process corner individually. This is highly time-consuming.

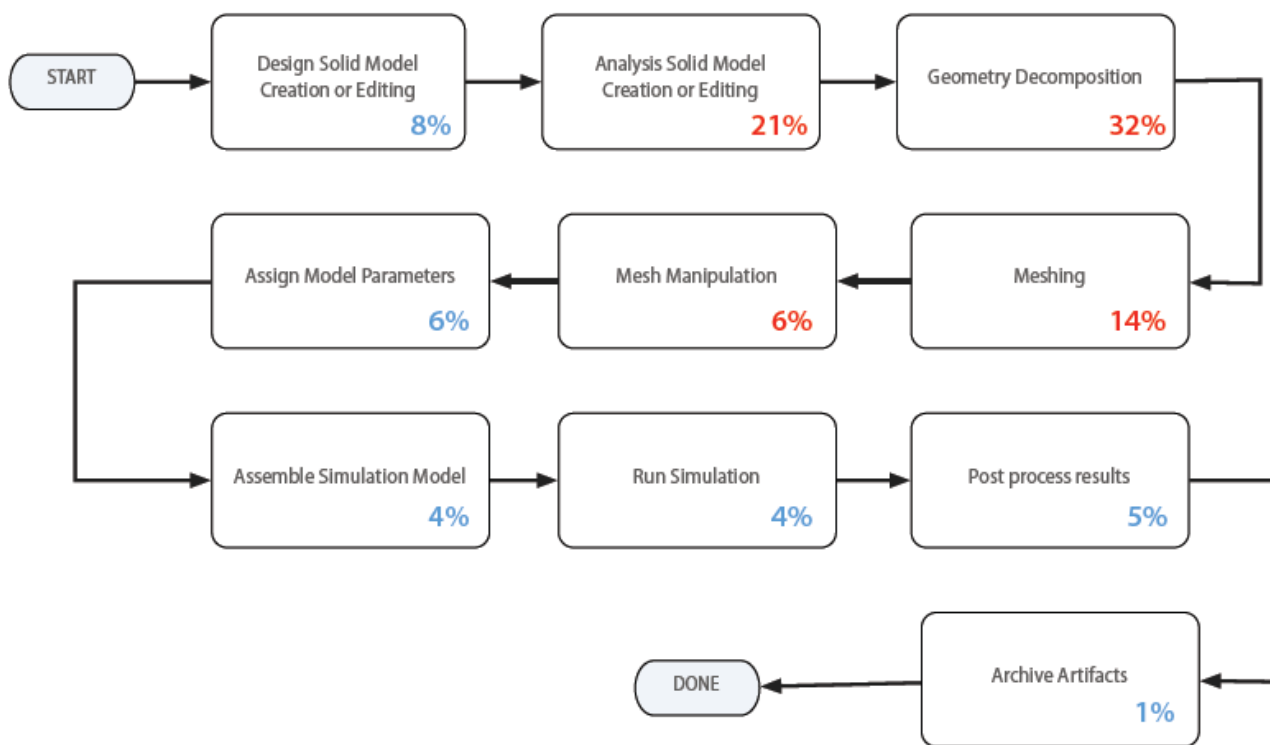


Рис. __. Design through analysis man hours spent (not CPU hours). A typical 10 step design through analysis cycle (excluding iterations) is shown. A staggering 73% of time is spent in geometry manipulation for meshing. [Courtesy: Matt Staten, Computing and Modeling Dept, Sandia National Labs]

Process level

•Process flow modeling: Process engineers still spend a significant amount of time hand-drawing process flows in Powerpoint, Illustrator or some such tool. Any change in the process flow requires starting afresh in compiling the flow diagrams. While 3D process visualization and virtual prototyping was introduced by IntelliSense in 1995, full 3D visualization is time-consuming for large MEMS devices. There is a need to obtain 2D cross sections of custom process flows with minimal computing overhead. Better yet, automatic assembly of Powerpoint presentations would greatly simplify the process flow modeling.

•Integrated cost and economics modeling: Cost and economics are often an afterthought during the design process. Any changes to the design, process, or package configurations requires re-running Excel spreadsheets. Trade-offs between various configurations become difficult to envision without years of expertise. At present, design and process engineers have little or no visibility into the cost equation. Empowering engineers to make squeezing gross margins out of the product.

•Deep Silicon Etch Simulations: Deep Reactive Ion Etching (DRIE) is here to become the main stay of MEMS process technology. First generation DRIE simulation tools such as RECIPE and RECIPE3D can be used to fine tune sidewall angle of structures locally. Manual mask corrections need to be applied to ensure uniform etching across the die. Next generation tools, similar to optical proximity correction tools for lithography, are needed to automatically correct the masks to ensure uniform local etching of devices.

•Incorporating accurate process details into meshes: While tools like FABViewer™ and MEMulator™ can produce realistic device renderings, FEA meshing is often based upon idealized geometries rather than true geometries. Better meshing algorithms are needed to automatically create realistic meshes that are efficient to use in FEA simulations.

Structural level (3D simulation)

•Enhanced ROMs: System model extraction techniques have improved over the years from simple linear macromodels to accurate non-linear reduced order models (ROMs) that can accurately capture stress stiffening, electrostatic spring softening, and fluid damping. With the advent of wafer-level packaging and demand for low-profile MEMS for mobile applications, more and more of the packaging functionality is being subsumed into the MEMS. Packaging effects like temperature/stress effects and shunt capacitance calculations need to be automatically included into the reduced order models.

•Parametric ROMs: While the use of high fidelity ROMs in system simulation has gained adoption, ROMs suffer from the disadvantage of being tied to the original geometry. In a sense they are black box models. Users may prefer to have control over some of the device parameters, such as material thickness or electrode separations, for systemlevel optimization scenarios. This necessitates the creation of parametric ROMs. Parametric ROMs can be derived using Latin Hyperspace (LHS) based design exploration use in system simulators.

Synthesis and Optimization tools

•Optimization tools: While most CAD/EDA tools provide links to third-party optimization tools

Лекція 9. Технічне забезпечення процесу проектування

such as ModeFrontier, iSight or ModelCenter, the adoption of these tools into the MEMS design flow is non-existent. Impediments include poor integration, a steep learning curve, and the lack of optimization training in the MEMS industry. Many of these impediments can be solved by tightly integrating these tools into the design workflow and significantly simplifying the optimization setup.

IntelliSense's Design Methodology

IntelliSense's software architecture is based upon a unique combination of the best of bottom-up process-driven design and topdown synthesis. Top-down methodology allows you to quickly explore a wide range of design options, while bottom-up design provides the accuracy to produce first-time-right silicon.

IntelliSense gives you a distinctive methodology to tackle the conversion of product specifications and requirements into a working product. The accurate bottom-up process-driven design and top-down schematic-driven synthesis are combined to get you to your designs faster and with less process iterations.

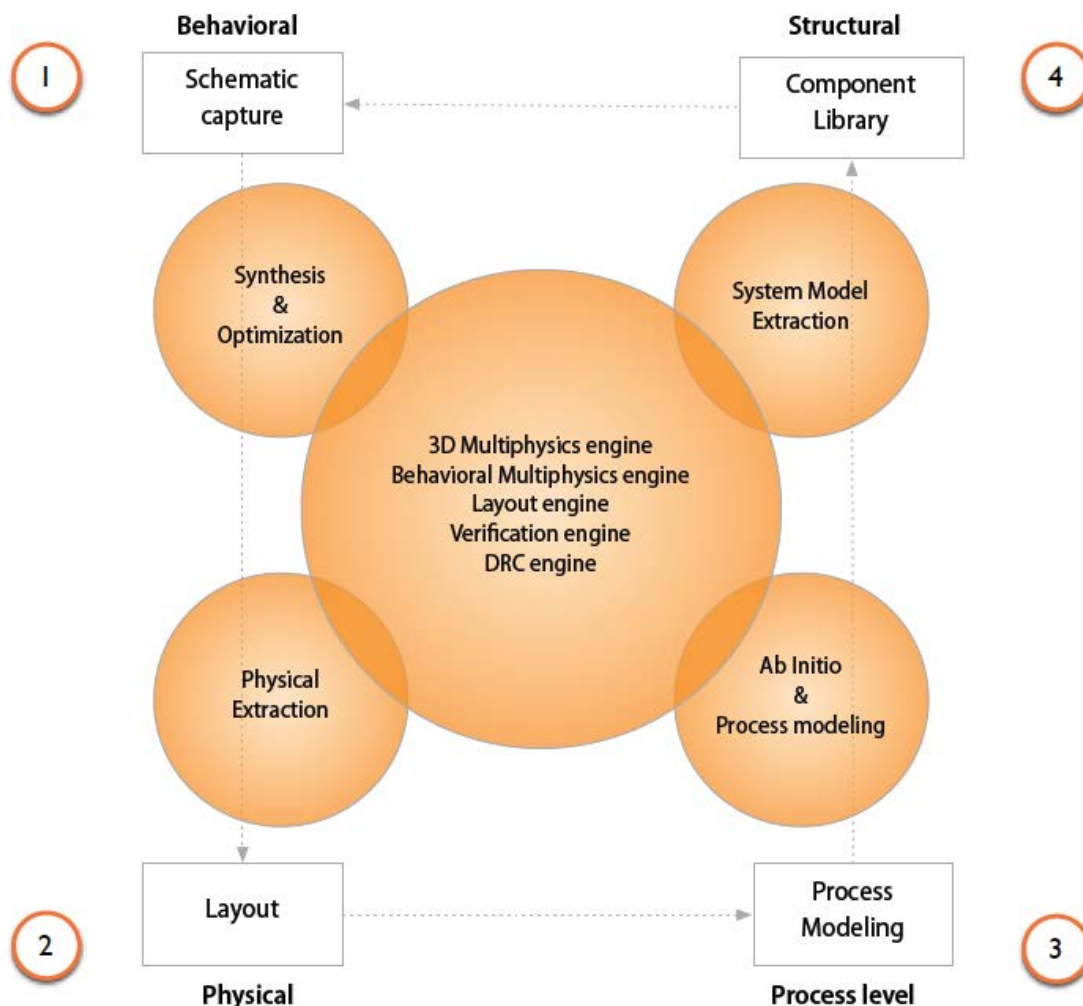


Рис. __. Design in IntelliSuite. IntelliSuite allows you to design and analyze your device at different levels.

Core computational engines and databases combined with synthesis, optimization, process modeling, physical and system model extraction provide a friction-free and efficient workflow.

СИСТЕМ

From the top down

State-of-the-art schematic capture and simulation tools allow you to take a hierarchical approach to the design space. SYNPLE provides a large multi-domain library of electrical, mechanical, thermal, digital and controls, and MEMS libraries. These elements may be combined in an effortless drag-and-drop fashion and then wired to create schematics of multi-scale, multi-domain systems. As a result, you can quickly survey a large design space before initiating a detailed analysis and verification process.

The top-down approach allows you to combine readily available component blocks into a netlist. Due to the simplified nature of the component models, users can perform device-level optimization using Design of Experiments (DoE), Robust Design or other techniques.

Users can start with component-based schematic capture and use optimization techniques to explore a vast design space. Built-in place and route algorithms can be then used to convert the schematic into a mask layout or an optimally meshed model ready for full 3D analysis.

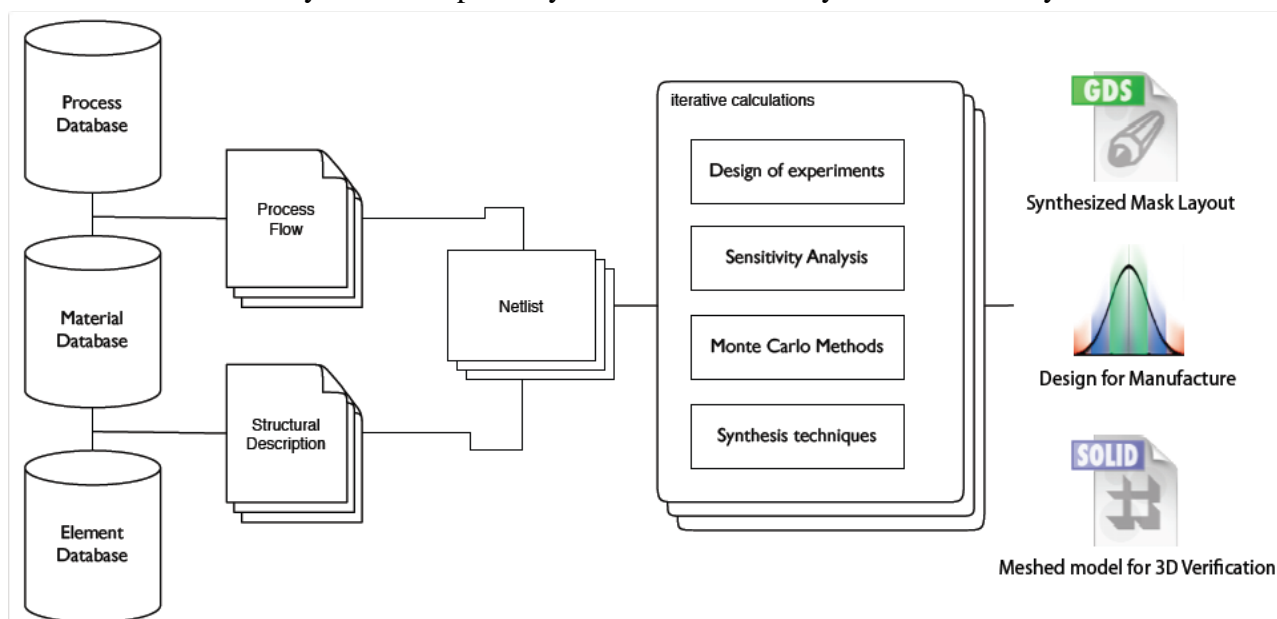


Рис. 1. Top-down modeling is fast but of lower fidelity. It can be used for rapid exploration of design space and optimizing the design for performance and manufacturability.

One step at a time

IntelliSuite's bottom up architecture is based upon process elements — familiar process steps, such as photolithography, thin film deposition, and selective etching form the basis of understanding the final device geometries. By systematically building the prototype in IntelliSuite, you can identify costly process bugs before entering the fab, which ultimately saves time and money. The process steps, combined with the mask geometries, can be used to build the final virtual device (power users can also import 3D geometries from popular CAD programs). In addition, the analysis modules (fullyintegrated thermo-electromechanical analysis, high-frequency electromagnetic analysis, microfluidics analysis) can be used to analyze the performance of MEMS models.

IntelliSuite features a comprehensive material and process database, allowing you to understand

Лекція 9. Технічне забезпечення процесу проектування

material properties like conductivity, film stresses and mechanical strength as a function of processing parameters. Subsequently, this enables you to produce more realistic models.

Etching has always been a bugbear in MEMS technology. We provide wet and dry etch simulators — a full anisotropic wet etch simulator for creating realistic models of your KOH, TMAH or EDP etches, and a dry etch simulator for simulating RIE/ICP and Bosch etch processes.

What's the bottom line? IntelliSuite allows you to use state-of-the-art model reduction techniques to automatically create compact system models from large finite element models.

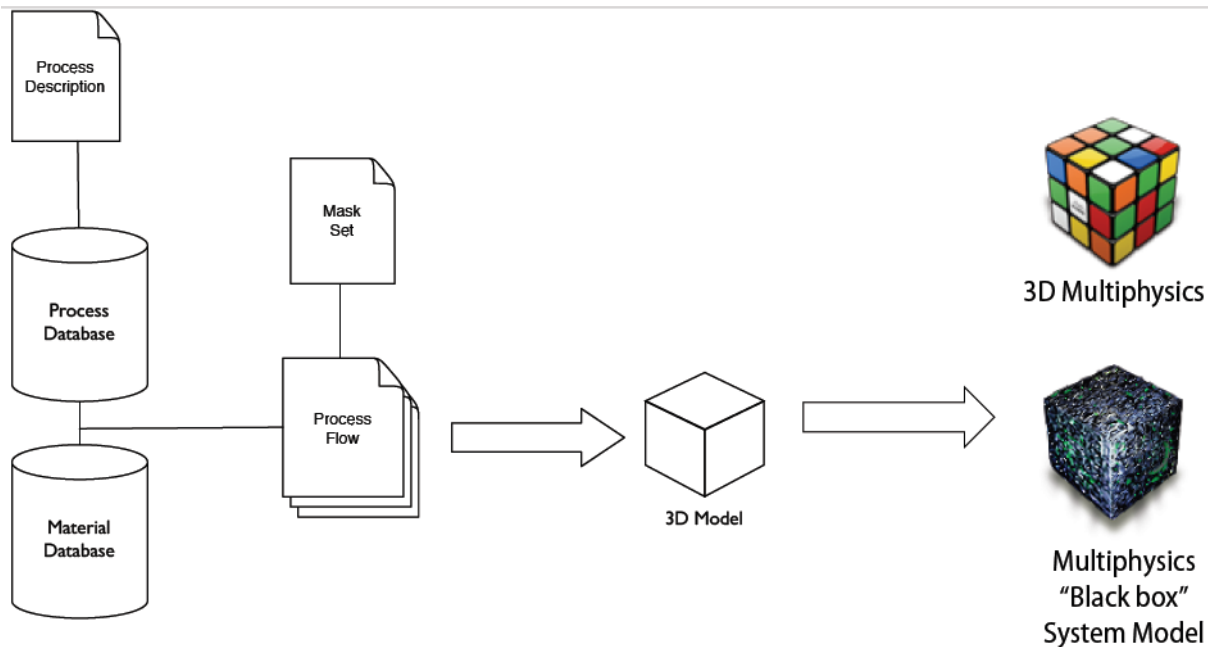


Рис. 9.1. Bottom-up modeling is accurate but slower. Use bottom-up modeling for accurately capturing the device behavior and encapsulating it into a black-box system model.

Closing the loop

IntelliSuite offers you a number of tools to close the loop between topdown and bottom-up modeling. Synthesis and placement tools such as MEMS-Synth and Hexpresso can automatically transform the schematic into a ready-to-use layout or a meshed structure for FEABEA analysis. In addition, graphics tools allow you to visualize the results of schematic level analysis in 3D, the natural context for MEMS design.

Similarly, System Model Extraction (SME) tools based upon energy storage and dissipation in multiple physical domains can accurately capture the dynamics of the MEMS device. The Reduced Order Models from SME can capture all of the device and packaging effects. The derived ROMs can be directly used in schematic-level co-simulation with the electronics or alternately exported into popular Hardware Description Languages (HDLs) for use in simulators such as PSPICE, HSPICE, Cadence Virtuoso, Mathworks Simulink, and MentorGraphics SystemVision.

By presenting a uniform framework for simultaneous top-down and bottom-up methodologies and toolsets to easily switch between methodologies, IntelliSuite allows information capture from the entire design team.

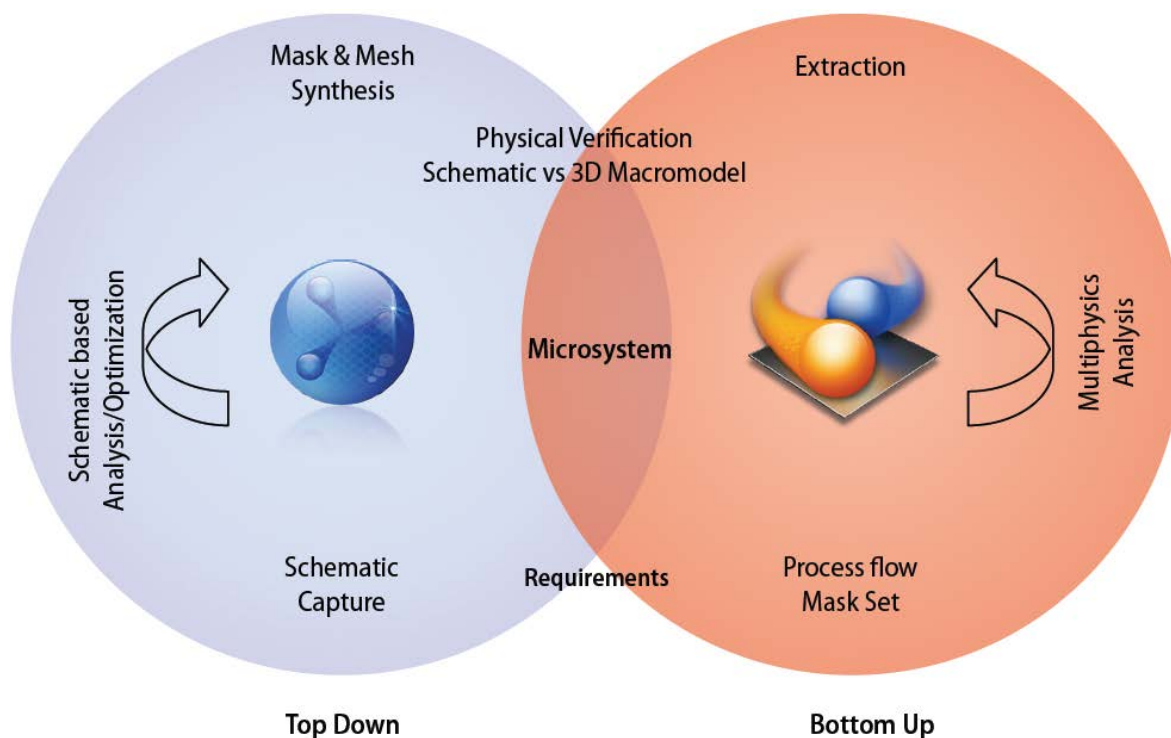


Рис. __. IntelliSuite allows you to combine the best of top down and bottom-up methodology into an efficient workflow.

Synthesis, optimization, and extraction tools are available to ensure an efficient workflow.

Final thoughts

Living Design Environment

Ultimately, the development of a new MEMS product succeeds or fails based on the communication (or lack thereof) between different functional teams. MEMS teams require seamless communication across design, processing, packaging, controls, readout electronics and increasingly, software.

While point tools can capture parts of the information, an integrated approach is needed to maintain and evolve the design across the product life cycle.

IntelliSuite is designed as a tightly integrated environment that will link your entire MEMS organization together. Built to scale from a point tool to an organization-wide tool, IntelliSuite unifies various engineering and manufacturing tasks into a single living design environment.

Used by MEMS professionals worldwide for design, development and manufacturing of MEMS, IntelliSuite has firmly established itself as an industry standard tool. IntelliSense's tools are now used by large and small organizations in over 30 countries, including more than 60% of the top MEMS companies.

ЗАГАЛЬНА ЛІТЕРАТУРА

- 1 Лебедєв, А. А. Введення в аналіз і синтез систем: Навчальний посібник / А. О. Лебедєв .- М.: МАІ, 2001 .- 352с.
- 2 Голдштейн, Г. Я. Інноваційний менеджмент / Г. Я. Гоштейн .- Таганрог: Видавництво ТРТУ, 1998 .- 132с.
- 3 Годін, Е. М., Харнайсов, К. З., Сокольський М. Л., Системи автоматизованого проектування та основи управління виробництвом: Навчальний посібник .- М.: МАІ, 2004 .- 680С.
- 4 Афанасьєва, О. В. Голік, Є. С. Первухін, Д. А. Теорія і практика моделювання складних технічних систем: Навчальний посібник / О. В. Афанасьєва, Є. С. Голік, Д. А. Первухін .- Спб: СЗТУ, 2005 .- 131с.
- 5 Chakrabarty K., Su F. Design Automation Challenges for Microfluidics-Based Biochips. - Montreux, Switzerland, 01-03 June 2005.
- 6 Zhang T., Chakrabarty K., Fair R. B. Microelectrofluidic Systems: Modeling and Simulation.- CRC Press, Boca Raton, FL, 2002.