

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
Інститут інформатики та радіоелектроніки

КОНСПЕКТ ЛЕКЦІЙ
з дисципліни
«САПР мікро- і наносистем»

для студентів спеціальності
172 «Телекомунікації та радіотехніка» (освітня програма
«Інтелектуальні технології мікросистемної радіоелектронної техніки»)
усіх форм навчання

Конспект лекцій з дисципліни "САПР мікро- і наносистем" для студентів спеціальності 172 «Телекомунікації та радіотехніка» (освітня програма «Інтелектуальні технології мікросистемної радіоелектронної техніки») усіх форм навчання / Уклад.: Фурманова Н.І., Фарафонов О.Ю., Малий О.Ю. – Запоріжжя: НУ «Запорізька політехніка», 2019. – 58 с.

Укладачі: Фурманова Наталія Іванівна, канд. техн. наук, доцент, доцент кафедри ІТЕЗ;
Фарафонов Олексій Юрійович, канд. техн. наук, доцент, доцент кафедри ІТЕЗ,
Малий Олександр Юрійович, канд. техн. наук, доцент кафедри ІТЕЗ

Рецензент: Миронова Наталя Олексіївна, канд. техн. наук, доцент каф. ПЗ

Відповідальний за випуск: Шило Галина Миколаївна, канд. техн. наук, доцент, зав. каф. ІТЕЗ

Розглянуто
на засіданні кафедри ІТЕЗ
протокол № 1 від 28.08.19 р.

Затверджено
на засіданні НМК ФРЕТ
протокол № 1 від 02.09.19 р.

ЗМІСТ

ВСТУП.....	5
1 Визначення, класифікація та інструментарій інформаційних технологій.....	6
2 Загальна характеристика процесів збору, передачі, обробки та зберігання інформації.....	8
3 Поняття «об'єктивність», «повнота», «достовірність», «адекватність», «доступність» і «актуальність інформації». Технологія забезпечення безпеки комп'ютерних систем. Проблеми використання інформаційних технологій	12
4 Інформаційна технологія підтримки прийняття рішень. Основні компоненти.....	15
5 Експертні системи та їх типи. Основні способи формалізованого представлення знань в нано- та мікросистемній техніці. Області застосування експертних систем	18
6 Автоматизовані системи наукових досліджень. Типова структура	23
7 Загальні відомості про системи автоматизованого проектування. Типова структура. Визначення і основне призначення CAD- / CAE- / CAM-систем	25
8 Основи автоматизації проектування мікроелектромеханічних і наноелектромеханічних систем. Методологія їх проектування	28
9 Інтегровані пакети автоматизованого проектування MEMS і НЕМС.....	30
10 Загальні відомості про комп'ютерне математичне моделювання. Класифікація математичних моделей. Імітаційне моделювання	33
11 Етапи, цілі та засоби комп'ютерного математичного моделювання	35
12 Особливості імітаційного моделювання виробничих систем.....	38
13 Основні параметри, що визначають перебіг і стан технологічного процесу виробництва компонентів нано- та мікросистемної техніки.....	40
14 Система моделей в технології нано- та мікросистемної техніки	42

15	Поняття про обчислювальний технологічний експеримент..	44
16	Загальна характеристика технології створення програмного забезпечення. Етапи розробки програмного забезпечення	47
17	Сучасні методи розробки ПЗ: метод спадного проектування, модульне проектування, CASE-технології.....	52

ВСТУП

Дисципліна **«САПР мікро- і наноструктур»** є частиною спеціальної підготовки інженера за спеціальністю «Телекомунікація та радіотехніка», освітня програма «Інтелектуальні технології мікросистемної радіоелектронної техніки». Даний конспект лекцій допоможе вивчити загальні характеристики процесів збору, передачі, обробки та зберігання інформації, організацію інформаційних процесів, інформаційні технології при виборі рішень в області технології нано- та мікросистемної техніки, організації, планування і управління виробництвом, а також автоматизовані системи наукових досліджень, системи автоматизованого проектування MEMS і NEMS, технології комп'ютерного моделювання нано- та мікросистемної техніки.

Ускладнення розроблюваних пристроїв мікро- і нанотехніки призводить до того, що підготовка фахівців з виробництва і проектування нано- та мікросистемної техніки повинна включати такі важливі складові формування базового набору компетенцій, як знання теоретичних основ інформаційних технологій, методів моделювання та проектування компонентів мікро- і нанотехніки, а також наявність практичних навичок використання сучасних засобів автоматизованого проектування MEMS і NEMS, моделювання наноструктур і нанотехнологій.

1 ВИЗНАЧЕННЯ, КЛАСИФІКАЦІЯ ТА ІНСТРУМЕНТАРІЙ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Інформаційна технологія (ІТ) - сукупність засобів і методів збору, обробки і передачі даних (первинної інформації) для отримання інформації нової якості про стан об'єкта, процесу або явища (інформаційного продукту).

Мета інформаційної технології – виробництво інформації для її аналізу людиною і прийняття на її основі рішення з виконання якої-небудь дії.

Глобальна інформаційна технологія включає моделі, методи і засоби, що формалізують і дозволяють використовувати інформаційні ресурси суспільства.

Базова інформаційна технологія призначена для певної області застосування (виробництво, наукові дослідження, навчання і т. д.).

Конкретні інформаційні технології реалізують обробку даних при вирішенні функціональних завдань користувачів (наприклад, завдання обліку, планування, аналізу).

На даний момент існують різні підходи до проблеми класифікації інформаційних технологій. Наведемо деякі з них. В одних роботах виділяють наступні види інформаційних технологій:

- інформаційна технологія обробки даних;
- інформаційна технологія управління;
- автоматизація офісу;
- інформаційна технологія підтримки прийняття рішень;
- інформаційна технологія експертних систем.

Інші джерела інформаційні технології поділяють на різні види наступним чином:

- функціонально-орієнтовані інформаційні технології, призначені для реалізації певних завдань;
- предметно орієнтовані інформаційні технології, призначені для вирішення конкретних завдань в певній предметній області;
- проблемно-орієнтовані інформаційні технології, призначені для вирішення типових прикладних задач.

Інструментарій інформаційної технології - один або кілька взаємопов'язаних програмних продуктів для певного типу комп'ютера, технологія роботи в якому дозволяє досягти поставленої користувачем

мети. Інформаційна система – людино-комп'ютерна система для підтримки прийняття рішень і виробництва інформаційних продуктів, що використовує комп'ютерну інформаційну технологію.

2 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПРОЦЕСІВ ЗБОРУ, ПЕРЕДАЧІ, ОБРОБКИ ТА ЗБЕРІГАННЯ ІНФОРМАЦІЇ

Збір інформації

Система збору інформації являє собою складний програмно-апаратний комплекс. Як правило, сучасні системи збору інформації не тільки забезпечують кодування інформації та її введення в ЕОМ, а й виконують попередню (первинну) обробку цієї інформації. **Збір інформації** - це процес отримання інформації з зовнішнього світу і приведення її до вигляду, стандартного для даної інформаційної системи.

Обмін інформацією між інформацією системою і навколишнім середовищем здійснюється за допомогою сигналів. Сигнал можна визначити як засіб перенесення інформації в просторі та часі. В якості носія сигналу в технології компонентів мікро- і нанотехніки можуть виступати електричний струм, світло, магнітне поле і т. д. Технічні системи для прийому сигналів оснащуються спеціальними пристроями.

Незалежно від носія інформації (сигналу) типовий процес обробки сигналу може бути охарактеризований наступними кроками.

На першому кроці вихідний (первинний) сигнал за допомогою спеціального пристрою (датчика) перетворюється в еквівалентний їй електричний сигнал (електричний струм).

На другому кроці вторинний (електричний) сигнал в певний виділений момент часу оцифровується спеціальним пристроєм - аналого-цифровим перетворювачем (АЦП). АЦП значенням електричного сигналу ставить у відповідність деяке число з кінцевого безлічі таких чисел. Таким чином, датчик і АЦП, пов'язані разом, складають цифровий вимірювальний пристрій. Якщо цей прилад оснастити деяким пристроєм для зберігання виміряної величини - регістром, - то на наступному кроці за командою від ЕОМ можна ввести це число в машину і піддавати потім будь-якій необхідній обробці.

Передача інформації

Взаємодія між територіально віддаленими об'єктами здійснюється за рахунок обміну даними. Доставка даних здійснюється за заданою адресою з використанням мереж передачі даних.

У сучасних умовах значного поширення набула розподілена обробка інформації, при цьому мережі передачі даних перетворюються в інформаційно-обчислювальні.

В якості найпростішого способу підвищення достовірності передачі інформації може використовуватися контроль на парність. Суть цього способу полягає в наступному. На вході в канал зв'язку проводиться підрахунок числа «1» в двійковій кодової послідовності - вхідному повідомленні. Якщо число «1» виявляється непарним, в хвіст переданого повідомлення додається «1», а якщо парним, то «0». На приймаючому кінці каналу зв'язку виконують аналогічний підрахунок, і якщо контрольна сума виявляється непарною, то робиться висновок про те, що при передачі відбулося спотворення інформації, в іншому випадку прийнята інформація визнається правильною (неспотвореною). В описаному способі використовується один додатковий контрольний розряд. Це дозволяє виявляти помилку передачі в разі спотворення одного-єдиного розряду в повідомленні. Цей дуже простий спосіб застосовують при передачі даних на великі відстані.

У тих випадках, коли ймовірність спотворення інформації при передачі велика, потрібні складніші методи. Так, перешкодостійкі коди дозволяють не тільки приймати рішення про правильність передачі інформації, але і в ряді випадків виправляти її. При контролі на парність єдиний спосіб отримати достовірну інформацію - повторна передача повідомлення. У разі коригувальних кодів, що дуже важливо при високій вартості передачі, є можливість виправляти помилки на приймаючому кінці каналу зв'язку, уникаючи, таким чином, повторної передачі інформації.

Обробка інформації.

Технологія електронної обробки інформації - людино-машинний процес виконання взаємопов'язаних операцій, що протікають у встановленій послідовності з метою перетворення вихідної (первинної) інформації в результуючу. Операція являє собою комплекс здійснюваних технологічних процесів, в результаті яких інформація перетворюється. Технологічні операції різноманітні за складністю, призначенням, техніці реалізації, виконуються на різному устаткуванні, багатьма виконавцями.

В умовах електронної обробки даних переважають операції,

що виконуються автоматично на машинах і пристроях, які зчитують дані, виконують операції за заданою програмою в автоматичному режимі за участю людини, або зберігаючи за користувачем функції контролю, аналізу і регулювання.

Побудова технологічного процесу виробництва компонентів мікро- і наносистемної техніки визначається наступними факторами: особливостями оброблюваної інформації, її обсягом, вимогами терміновості і точності обробки, типами, кількістю і характеристиками застосовуваних технічних засобів. Вони лягають в основу організації технології, яка включає встановлення переліку, послідовності і способів виконання операцій, порядку роботи фахівців і засобів автоматизації, організацію робочих місць, встановлення тимчасових регламентів взаємодії і т. д.

Організація технологічного процесу повинна забезпечити його економічність, комплексність, надійність функціонування, високу якість робіт. Це досягається використанням системотехнічного підходу до проектування технології. При цьому має місце комплексний взаємозалежний розгляд всіх чинників, шляхів, методів побудови технології, застосування елементів типізації та стандартизації, а також уніфікації схем технологічних процесів.

Розрізняють два основних типи організації технологічних процесів:

- предметний;
- поопераційний.

Предметний тип організації технології передбачає створення паралельно діючих технологічних ліній, що спеціалізуються на обробці інформації та вирішенні конкретних комплексів завдань і організують поопераційний обробку даних усередині лінії.

Поопераційний (потоківий) тип побудови технологічного процесу передбачає послідовне перетворення оброблюваної інформації відповідно до технології, представлені у вигляді безперервної послідовності змінюють один одного операцій, що виконуються в автоматичному режимі.

Зберігання інформації викликане багаторазовим її використанням, застосуванням постійної інформації, необхідністю комплектації первинних даних до їх обробки. Здійснюється на машинних носіях у вигляді інформаційних масивів, де дані розташовуються за встановленими в процесі проектування ознаками

групування.

Контроль безпеки даних і систем підрозділяється на контроль достовірності даних, безпеки даних і комп'ютерних систем.

Контроль достовірності даних виконується програмно під час введення та обробки. Засоби безпеки даних і програм захищають їх від копіювання, спотворення, несанкціонованого доступу. Засоби безпеки комп'ютерних систем забезпечують захист від крадіжки, вірусів, неправильної роботи користувачів, несанкціонованого доступу.

Контроль достовірності даних

Основне завдання полягає у визначенні місця розташування спотворених даних та їх виправлення.

Розрізняють три типи контролю достовірності:

- синтаксичний;
- семантичний;
- прагматичний.

Синтаксичний контроль забезпечує перевірку типу полів, наявність заборонених символів, порядку проходження реквізитів. Виконується на основі опису вхідних документів.

Семантичний контроль перевіряє логічні взаємозв'язки значень реквізитів, несуперечливість даних та їх узгодженість.

Прагматичний контроль перевіряє щільність, своєчасність, повноту даних, що надаються для прийняття рішень. Виконується на основі вихідних документів.

3 ПОНЯТТЯ «ОБ'ЄКТИВНІСТЬ», «ПОВНОТА», «ДОСТОВІРНІСТЬ», «АДЕКВАТНІСТЬ», «ДОСТУПНІСТЬ» І «АКТУАЛЬНІСТЬ ІНФОРМАЦІЙ». ТЕХНОЛОГІЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ КОМП'ЮТЕРНИХ СИСТЕМ. ПРОБЛЕМИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Об'єктивність і суб'єктивність інформації

Поняття об'єктивності є відносним, так як методи є суб'єктивними. Більш об'єктивною прийнято вважати ту інформацію, в яку методи вносять менший суб'єктивний елемент.

Повнота інформації

Повнота інформації багато в чому характеризує якість інформації і визначає достатність даних для прийняття рішень або для створення нових даних на основі наявних. Чим повніші дані, тим більш широкий діапазон методів, які можна використовувати.

Достовірність інформації

Дані виникають у момент реєстрації сигналів, але не всі сигнали є «корисними» - завжди присутній якийсь рівень сторонніх сигналів, в результаті чого корисні дані супроводжуються певним рівнем «інформаційного шуму». Якщо корисний сигнал зареєстрований більш чітко, ніж сторонні сигнали, достовірність інформації може бути вищою.

Адекватність інформації - це ступінь відповідності реальному об'єктивному стану справи. Неадекватна інформація може утворюватися при створенні нової інформації на основі неповних або недостовірних даних. Однак і повні, і достовірні дані можуть призводити до створення неадекватної інформації в разі застосування до них неадекватних методів.

Доступність інформації - міра можливості отримати ту чи іншу інформацію. На ступінь доступності інформації впливають одночасно як доступність даних, так і доступність адекватних методів для їх інтерпретації. Відсутність доступу до даних або відсутність адекватних методів обробки даних призводить до однакового результату: інформація виявляється недоступною.

Актуальність інформації - це ступінь відповідності інформації поточного моменту часу. Оскільки інформаційні процеси розтягнуті в часі, то достовірна і адекватна, але застаріла інформація

може приводити до помилкових рішень.

Репрезентативність інформації пов'язана з правильністю її відбору і формування з метою адекватного відображення властивостей об'єкта. Важливе значення тут мають:

- правильність концепції, на базі якої сформульовано вихідне поняття;
- обґрунтування відбору істотних ознак і зв'язків відображуваного явища.

Змістовність інформації відображає семантичну ємність, рівну відношенню кількості семантичної інформації в повідомленні до обсягу оброблюваних даних.

Точність інформації визначається ступенем близькості одержуваної інформації до реального стану об'єкта, процесу, явища і т. д.

Стійкість інформації відображає її здатність реагувати на зміни вихідних даних без порушення необхідної точності. Стійкість інформації, як і репрезентативність, обумовлена обраною методикою її відбору і формування.

Технологія забезпечення безпеки комп'ютерних систем

Крадіжка програмних засобів, порушення їх захисту, псування програмних продуктів з'явилися одночасно з виникненням комп'ютерних технологій. Але найбільшу увагу до цих питань привернув запуск в жовтні 1988 р. Робертом Морісом вірусу в комп'ютерну мережу Arpanet. В результаті був повністю або частково заблокований ряд загальнонаціональних комп'ютерних мереж, зокрема Internet, C&net, Bitnet. Загальний збиток оцінюється фахівцями в 100 млн дол. Тому питаннями безпеки комп'ютерних систем з моменту їх виникнення займалися відомства з охорони державних і військових таємниць, а в даний час - законодавчі служби та інститути.

Під час експлуатації найбільшої шкоди завдають віруси. На великих ЕОМ команди обробки переривань були привілейованими, тобто недоступними програмісту. Тому на великих ЕОМ віруси не приносили істотних збитків. Переривання на ПЕОМ обробляються непривілейованими командами і можуть бути доступними як операційній системі, так і прикладним програмам.

Прикладні програми можуть перехопити переривання, виконати будь-які дії і повернути управління. Так утворюються віруси, які по середовищі розмноження поділяються:

- на файлові, що заражають файли з розширенням;
- boot-віруси, що заражають сектор початкового завантаження;
- компільовані, що змінюють текст вихідних модулів;
- мережеві, які запускають віруси по мережі і є найбільш руйнівними.

Віруси можуть заражати програми, диски, операційні системи. Для боротьби з ними розроблені програми, які поділяються на такі види:

1. Детектори і ревізори - перевіряють цілісність інформації. В основному вони запам'ятовують довжину програми або файлу, підраховують і перевіряють контрольні суми програм, файлів. Є найбільш перспективними для виявлення нових типів вірусів.

2. Програми-фаги - вирізають текст, що повторюється, або віруси.

3. Програми-фільтри - стежать за несанкціонованими діями з обробки переривань.

Проблеми використання інформаційних технологій:

1. Інформаційний потік перевершує обмежені можливості людини щодо сприйняття і переробки інформації.

2. Виникає велика кількість надлишкової інформації (так званий «інформаційний шум»), яка ускладнює сприйняття корисної для споживача інформації.

3. Виникають економічні, політичні та інші бар'єри, які перешкоджають поширенню інформації (наприклад, через секретність).

Частковий вихід з інформаційної кризи полягає в застосуванні нових інформаційних технологій. Впровадження сучасних засобів і методів зберігання, обробки та передачі інформації багаторазово знижують бар'єр доступу до неї і швидкість пошуку. Зрозуміло, одні лише технології не можуть вирішити проблему, що має і економічний характер (інформація коштує грошей), і юридичний (інформація має власника), і ряд інших. Ця проблема комплексна і вирішується зусиллями як кожної країни, так і світової спільноти в цілому.

4 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ. ОСНОВНІ КОМПОНЕНТИ

Системи підтримки прийняття рішень і відповідна їм інформаційна технологія з'явилися в кінці 70-х-початку 80-х рр. ХХ ст., чому сприяли широке поширення персональних комп'ютерів, стандартних пакетів прикладних програм, а також успіхи в створенні систем штучного інтелекту.

Головною особливістю інформаційної технології підтримки прийняття рішень є якісно новий метод організації взаємодії людини і комп'ютера. Прийняття рішення, що є основною метою цієї технології, відбувається в результаті ітераційного процесу, в якому беруть участь:

- система підтримки прийняття рішень (СППР) в ролі обчислювального ланки і об'єкта управління;
- особа, яка приймає рішення, що оцінює отриманий результат обчислень на комп'ютері.

Закінчення ітераційного процесу відбувається з волі людини. В цьому випадку можна говорити про здатність інформаційної системи спільно з користувачем створювати нову інформацію для прийняття рішень.

Додатково до цієї особливості інформаційної технології підтримки прийняття рішень можна вказати ще ряд її відмінних характеристик:

- орієнтація на рішення погано структурованих формалізованих завдань;
- поєднання традиційних методів доступу і обробки комп'ютерних даних з можливостями математичних моделей і методами вирішення завдань на їх основі;
- спрямованість на непрофесійного користувача комп'ютера;
- висока адаптивність, що забезпечує можливість пристосовуватися до особливостей наявного технічного і програмного забезпечення, а також вимогам користувача.

Інформаційна технологія підтримки прийняття рішень може використовуватися на будь-якому рівні управління. Крім того, рішення, що приймаються на різних рівнях управління, часто повинні координуватися. Тому важливою функцією і систем, і технологій є координація осіб, які приймають рішення, як на різних рівнях управління, так і на одному рівні.

Основні компоненти

Розглянемо структуру системи підтримки прийняття рішень (рис. 4.1), а також функції складових її блоків, які визначають основні технологічні операції.

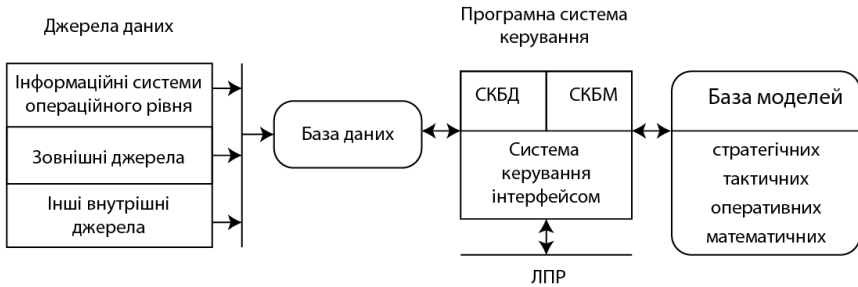


Рисунок 4.1 – Основні компоненти інформаційної технології підтримки прийняття рішень

До складу системи підтримки прийняття рішень входять три головні компоненти: база даних, база моделей і програмна підсистема, яка складається з системи управління базою даних (СКБД), системи управління базою моделей (СУБМ) і системи управління інтерфейсом між користувачем і комп'ютером. База даних відіграє в інформаційній технології підтримки прийняття рішень важливу роль. Дані можуть використовуватися безпосередньо користувачем для розрахунків за допомогою математичних моделей.

Розглянемо джерела даних і їх особливості:

1. Частина даних надходить від інформаційної системи операційного рівня. Щоб використовувати їх ефективно, дані повинні бути попередньо оброблені.

2. Крім даних про операції підприємства для функціонування системи підтримки прийняття рішень потрібні і інші внутрішні дані, наприклад, дані про рух персоналу, інженерні дані і т. п., які повинні бути своєчасно зібрані, введені і підтримані.

3. Важливе значення, особливо для підтримки прийняття рішень на верхніх рівнях управління, мають дані із зовнішніх джерел. На відміну від внутрішніх даних, зовнішні зазвичай купуються в організацій, що спеціалізуються на їх зборі.

4. В даний час широко досліджується питання про включення в базу даних ще одного джерела даних - документів, що включають в себе записи, листи, контракти, накази і т. д. Якщо вміст цих документів буде записано і потім оброблено за деякими ключовими характеристиками (постачальникам, споживачам, дат, видам послуг і ін.), то система отримає новий потужний джерело інформації.

База моделей.

Метою створення моделей є опис і оптимізація деякого об'єкта або процесу. Використання моделей забезпечує проведення аналізу в системах підтримки прийняття рішень. Моделі, базуючись на математичній інтерпретації проблеми, за допомогою певних алгоритмів сприяють знаходженню інформації, корисної для прийняття правильних рішень.

Приклад. Модель лінійного програмування дає можливість визначити найбільш вигідну виробничу програму випуску декількох видів продукції при заданих обмеженнях на ресурси.

5 ЕКСПЕРТНІ СИСТЕМИ ТА ЇХ ТИПИ. ОСНОВНІ СПОСОБИ ФОРМАЛІЗОВАНОГО ПРЕДСТАВЛЕННЯ ЗНАНЬ В НАНО- ТА МІКРОСИСТЕМНІЙ ТЕХНІЦІ. ОБЛАСТІ ЗАСТОСУВАННЯ ЕКСПЕРТНИХ СИСТЕМ

Найбільший прогрес серед комп'ютерних інформаційних систем відзначений області розробки експертних систем (ЕС), заснованих на використанні елементів штучного інтелекту. Експертні системи дають можливість фахівцеві отримувати консультації експертів з будь-яких проблем, на основі яких цими системами накопичені знання.

Під штучним інтелектом (ШІ) зазвичай розуміють здатності комп'ютерних систем до таких дій, які називалися б інтелектуальними, якби виходили від людини. Найчастіше тут маються на увазі здібності, пов'язані з людським мисленням. Роботи в області штучного інтелекту не обмежуються експертними системами. Вони також включають в себе створення роботів, систем, що моделюють нервову систему людини, його слух, зір, нюх, здатність до навчання. Рішення спеціальних завдань вимагає спеціальних знань.

Головна ідея використання технології експертних систем полягає в тому, щоб отримати від експерта його знання і, завантаживши їх у пам'ять комп'ютера, використовувати кожного разу, коли в цьому виникне необхідність.

Будучи одним з основних додатків штучного інтелекту, експертні системи є комп'ютерні програми, які трансформують досвід експертів в якій-небудь області знань в форму евристичних правил.

На практиці ЕС використовуються перш за все, як системи-порадники в тих ситуаціях, де фахівець сумнівається у виборі правильного рішення. Експертні знання, що зберігаються в пам'яті системи, більш глибокі і повні, ніж відповідні знання користувача. ЕС знаходять поширення при вирішенні завдань з прийняттям рішень в умовах невизначеності (неповноти) для розпізнавання образів, в прогнозуванні, діагностиці, плануванні, управлінні, конструюванні і т.д.

Типова експертна система складається з розв'язувача (інтерпретатора), БД (бази даних), БЗ (бази знань), компонентів придбання знань, пояснювального і діалогового компонентів.

БД призначена для зберігання вихідних і проміжних даних, що використовуються для вирішення завдань, фактографічних даних.

Розв'язувач, використовуючи вихідні дані з БД і БЗ, забезпечує вирішення завдань для конкретних ситуацій.

Компонент придбання знань автоматизує процес наповнення БЗ.

Пояснювальний компонент пояснює, як система отримала розв'язок задачі (або чому не отримала) і які знання вона при цьому використовувала. Діалоговий компонент забезпечує діалог між експертною системою і користувачем в процесі виконання завдання і придбання знань.

Експертні системи створюються для вирішення різного роду завдань професійної діяльності людини і в залежності від цього виконують різні функції.

Типи експертних систем

1. Експертні системи першого покоління

Призначені для розв'язання добре структурованих задач, що вимагають невеликого обсягу емпіричних знань. Сюди відносяться класифікаційні завдання і завдання вибору з наявного набору варіантів.

2. Оболонки ЕС

Мають механізм введення-виведення, але БЗ порожня. Потрібне налаштування на конкретну предметну область. Знання купуються в процесі функціонування ЕС, здатної до самонавчання.

3. Гібридні ЕС

Призначені для вирішення різних завдань з використанням БЗ. Це завдання з використанням методів системного аналізу, дослідження операцій, математичної статистики, обробки інформації. Користувач має доступ до об'єктивізованих знань, що містяться в БЗ і пакетах прикладних програм.

4. Мережеві ЕС

Між собою пов'язані кілька експертних систем. Результати вирішення однієї з них є вихідними даними для іншої системи. Ефективні при розподіленій обробці інформації.

При розробці експертних систем повинні брати участь:

- експерт тієї предметної області, завдання якої буде вирішувати система;

- інженер по знаннях - фахівець з розробок систем;
- програміст - фахівець з розробки інструментальних засобів.

Експерт визначає знання, тобто описує предметну область у вигляді сукупності даних і правил, забезпечує повноту і правильність введених в експертну систему знань. Дані визначають об'єкти, їх характеристики і значення. Правила вказують на способи маніпулювання даними.

Інженер по знаннях допомагає експерту: виявити і структурувати знання, необхідні для функціонування експертної системи; здійснити вибір інструментальних засобів, які найбільш ефективні для вирішення завдань в даній предметній області; вказати способи представлення знань.

Програміст розробляє інструментальну середу, що включає всі компоненти експертної системи, виробляє її сполучення з іншими існуючими системами.

Основні способи формалізованого представлення знань в нано- та мікросистемній техніці

1. Подання знань продукційними правилами

Продукційні правила представляють знання в формі ЯКЩО-ТО. Системи, що використовують уявлення знань продукційними правилами, називаються продукційними. Це найбільш наочний і простий спосіб. У таких системах подання знань є засоби, що дозволяють використовувати в даних і правилах нечітку інформацію з певною ймовірністю, званої фактором впевненості.

2. Логіка предикатів (розділ математичної логіки)

Константи і змінні визначають окремі об'єкти і позначаються буквами або набором літер (U, V, W, X, Y). Послідовність з n констант або змінних (зазвичай $n > 1$) називається функцією. Атомарним предикатом називається послідовність з n сутностей і понять, описаних константами, змінними або функціями. Предикат приймає одне з двох значень: істина або брехня. Предикат, в якому всі змінні, константи і функції пов'язані між собою, називається пропозицією. Пропозиції використовуються для представлення знань.

Логіка предикатів забезпечує високий рівень модульності знань (представляє їх як єдине ціле в певній предметній області) і дозволяє з'ясувати, наявні чи відсутні протиріччя між новими і вже існуючими знаннями. Але надмірний рівень формалізації

представлення знань, складність їх прочитання знижують ефективність обробки. Крім цього, в логіці предикатів все відносини описуються предикатами, що не дозволяє при комп'ютерній обробці повністю відобразити властивості структури даних.

3. Семантичні мережі

Знання можна розглядати як відносини між поняттями і сутностями, які є конкретними об'єктами реального світу. Поняття і відносини можна представити у вигляді семантичної мережі, що складається з вершин і дуг. У вершинах розташовуються поняття, а спрямовані зв'язку між вершинами відповідають різного роду відносин між цими поняттями.

Семантичні мережі можуть бути виконані учнями, що означає можливість автоматичного додавання в мережу нових вузлів у міру появи в досвіді її використання нових понять, а також збільшення вагових коефіцієнтів, відповідних дуг. У процесі її навчання між існуючими вузлами також можуть встановлюватися додаткові зв'язку.

4. Фреймові системи

Фрейми розглядаються як структура опису окремої сутності або поняття. Вони можуть бути у вигляді їх сукупностей, які подаються як окреме безліч знань, що відносяться до одного об'єкту. Кожен фрейм складається з безлічі елементів, які називаються слотами, які в свою чергу представляються певною структурою даних. Кожен фрейм і слот мають ім'я, єдине у всій фреймовій системі. У значенні слот містить конкретну інформацію.

Області застосування ЕС

ЕС в задачах інтерпретації, як правило, використовують інформацію від датчиків для опису ситуації. Як приклад наведемо інтерпретацію показань вимірювальних приладів на підприємстві для визначення стану технологічного процесу. Інтерпретуючі системи мають справу не з чіткими символічними уявленнями проблемної ситуації, а безпосередньо з реальними даними. Вони стикаються з труднощами, яких немає у систем інших типів, тому що їм доводиться обробляти інформацію зашумлену, недостатню, неповну, ненадійну або помилкову. Їм необхідні спеціальні методи реєстрації характеристик безперервних потоків даних, сигналів або зображень і методи їх символічного уявлення.

ЕС в задачах прогнозування визначають ймовірні наслідки заданих ситуацій. Системи прогнозування іноді використовують імітаційне моделювання, тобто програми, які відображають причинно-наслідкові взаємозв'язки в реальному світі, щоб згенерувати ситуації або сценарії, які можуть виникнути при тих чи інших вхідних даних. Ці можливі ситуації разом зі знаннями про процеси, що породжують ці ситуації, утворюють передумови для прогнозування.

ЕС в задачах діагностики використовують опису ситуацій, характеристики поведінки або знання про конструкції компонента, щоб встановити ймовірні причини неправильного функціонування діагностується системи. Приклад: локалізація несправностей в електронних приладах.

ЕС, що застосовуються в галузі проектування, розробляють конфігурації об'єктів з урахуванням набору обмежень, властивих проблемі. З огляду на те, що проектування настільки тісно пов'язане з плануванням, багато проектувальних систем містять механізми розробки і уточнення планів для досягнення бажаного проекту. Найбільш часто зустрічаються області застосування плануючих ЕС: хімія, електроніка та військова справа.

ЕС, які використовуються для вирішення завдань спостереження, порівнюють дійсну поведінку з очікуваною поведінкою системи. Прикладом може служити стеження за показами вимірювальних приладів в ядерних реакторах з метою виявлення аварійних ситуацій.

Спостерігаючи ЕС за самою своєю природою повинні працювати в режимі реального часу. ЕС в задачах ремонту апаратури слідує плану, який наказує деякі рецепти відновлення. Прикладом є налагодження мас-спектрометра, тобто установка ручок регулювання приладу в положення, що забезпечує досягнення оптимальної чутливості, сумісної з правильним ставленням величин піків і їх форми.

ЕС в області навчання піддають діагностиці, «налагодженні» і виправлення («ремонт») поведінка учня. Як приклад наведемо навчання студентів пошуку несправностей в електричних ланцюгах.

6 АВТОМАТИЗОВАНІ СИСТЕМИ НАУКОВИХ ДОСЛІДЖЕНЬ. ТИПОВА СТРУКТУРА

Автоматизовані системи наукових досліджень (АСНД) представляють собою програмно-апаратні комплекси, які обробляють дані, що надходять від різного роду експериментальних установок і вимірювальних приладів, і на основі їх аналізу полегшують виявлення нових ефектів і закономірностей (рис. 6.1).



Рисунок 6.1 - Типова структура АСНД

Блок зв'язку з вимірювальною апаратурою перетворює до потрібного вигляду інформацію, що надходить від вимірювальної апаратури. У базі даних зберігається інформація, що надійшла з блоку зв'язку з вимірювальною апаратурою, а також заздалегідь введена з метою забезпечення працездатності системи. Розрахунковий блок, виконуючи програми з пакета прикладних програм, виробляє всі математичні розрахунки, в яких може виникнути потреба в ході наукових досліджень. Розрахунки можуть виконуватися як на вимогу дослідника, так і блоку імітаційного моделювання. При цьому на

основі математичних моделей відтворюється процес, що відбувається в зовнішньому середовищі.

Експертна система моделює міркування фахівців даної предметної області. З її допомогою дослідник може класифікувати спостерігаються явища, діагностувати протягом досліджуваних процесів. АСНД набули широкого поширення в молекулярної хімії, мінералогії, біохімії, фізиці елементарних частинок та інших науках.

7 ЗАГАЛЬНІ ВІДОМОСТІ ПРО СИСТЕМИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ. ТИПОВА СТРУКТУРА. ВИЗНАЧЕННЯ І ОСНОВНЕ ПРИЗНАЧЕННЯ CAD- / CAE- / CAM-СИСТЕМ

Близькими за своєю структурою і функціями до систем автоматизації наукових досліджень виявляються системи автоматизованого проектування (САПР).

САПР - комплекс програмних і апаратних засобів, призначених для автоматизації процесу проектування людиною технічних виробів або продуктів інтелектуальної діяльності.

Проектування нових виробів - основне завдання винахідників-конструкторів - протікає в кілька етапів, таких як нормування задуму, пошук фізичних принципів, що забезпечують реалізацію задумів і необхідні значення конструкції, пошук конструктивних рішень, їх розрахунок і обґрунтування, створення дослідного зразка, розробка технологій промислового виготовлення.

Якщо формування задуму і пошук фізичних принципів поки залишаються чисто творчими, не піддаються автоматизації етапами, то при конструюванні і розрахунках з успіхом можуть бути застосовані САПР (рис. 1.7.1). База даних, блок імітаційного моделювання, розрахунковий блок і експертна система виконують функції, аналогічні функціям відповідних блоків АСНД. Замість блоку зв'язку з вимірювальною апаратурою в САПР є блок формування завдань. Проектувальник вводить в блок технічне завдання на проектування із зазначенням цілей, які необхідно досягти при проектуванні, і всі обмеження, які не можна порушити.

Блок підготовки технічної документації полегшує її створення для подальшого виготовлення виробу. Апаратне забезпечення САПР становить ЕОМ з набором пристроїв, необхідних для введення і виведення графічної інформації (графічний пристрій, світлове перо, графічний планшет та ін.).

Системи автоматизації проектування включають в себе системи інженерної графіки (CAD), системи інженерних розрахунків (CAE), системи автоматизації підготовки та управління виробництва (CAM). CAD-системи призначені для вирішення конструкторських задач і оформлення конструкторської документації. В сучасні CAD-системи входять модулі моделювання тривимірної об'ємної

конструкції і оформлення креслень і текстової конструкторської документації (специфікацій, відомостей і т. д.), показано на рис. 7.1.

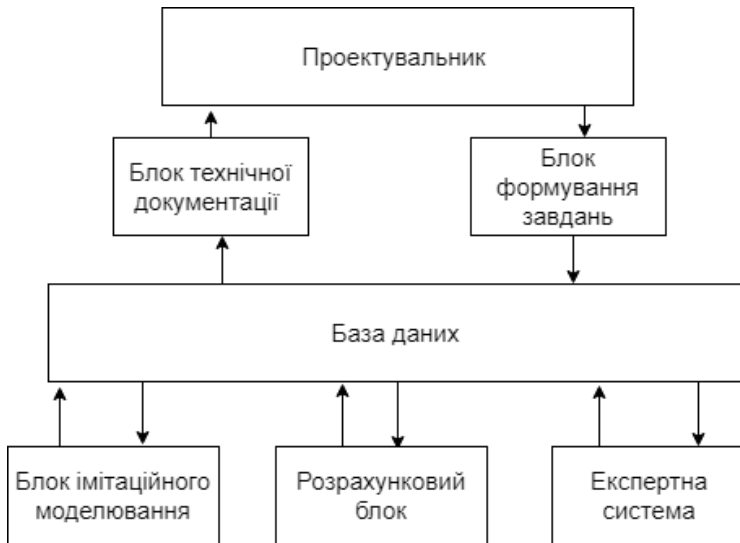


Рисунок 7.1 - Типова схема САПР

CAE-системи - це клас систем, кожна з яких дозволяє вирішувати певну розрахункову задачу, починаючи від розрахунків на міцність, аналізу і моделювання теплових процесів до розрахунків гідравлічних систем і машин, розрахунків процесів лиття.

У CAE-системах також використовується тривимірна модель виробу, створена в системі CAD.

CAM-системи призначені для проектування обробки виробів на верстатах з числовим програмним управлінням (ЧПУ) і видачі програм для цих верстатів. В даний час CAM-системи є одним з основних способів виготовлення складнопрофільних деталей і скорочення циклу їх виробництва. У них використовується тривимірна модель деталі, створена в CAD-системі.

Традиційно проводиться умовний розподіл CAD- / CAM- / CAE-систем на системи верхнього (наприклад, Unigraphics), середнього (наприклад, Solid Edge) і нижнього рівня (наприклад, AutoCAD).

Основні критерії вибору систем:

- функціональні можливості;
- наявність унікальних функцій;
- вартість;
- простота інтерфейсу і легкість навчання.

8 ОСНОВИ АВТОМАТИЗАЦІЇ ПРОЕКТУВАННЯ МІКРОЕЛЕКТРОМЕХАНІЧНИХ І НАНОЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ. МЕТОДОЛОГІЯ ЇХ ПРОЕКТУВАННЯ

Мікроелектромеханічні системи (МЕМС) - це інтегровані системи, що комбінують електричні і механічні компоненти, виготовлені за технологіями, сумісним з технологією ІС, і мають розміри від 0,1 до 10 см.

Наноелектромеханічні системи (НЕМС) - це інтегровані системи, що комбінують електричні і механічні компоненти, виготовлені за технологіями, сумісним з технологією ІС, і мають розміри від 0,1 до 100 нм.

У загальному випадку вони забезпечують конвертування величин різної фізичної природи (електричної, механічної і т. д.). У зв'язку з цим узагальнена функціональна блок-схема МЕМС включає в себе наступні компоненти:

- актюатори і / або сенсори;
- системи випромінювання (антени);
- ланцюги, які використовуються для обробки інформації від сенсорів, актюаторів;
- ланцюги управління.

Актюатор (складова частина МЕМС) - це пристрій, який перетворює енергію в керований рух. Завдяки своїй простоті найбільшого поширення в МЕМС отримали електростатичні перетворювачі. Електростатична активація застосовується приблизно в одній третині актюаторів.

Сенсор - це пристрій, що перетворює вхідний вплив різної фізичної природи в електричний сигнал.

Методи проектування систем, які базуються на використанні електроніки та точної механіки, виготовлених за технологіями, сумісними з технологією виробництва інтегральних схем, істотно відрізняються від раніше застосовуваних методів проектування електромеханічних систем. В основі маршрутів проектування МЕМС і НЕМС лежить міждисциплінарний підхід, що дозволяє врахувати різноманітні фізичні взаємодії між різними областями системи: електричної, механічної, теплової та ін. Важливою проблемою в наноінженерії є вибір моделей, які адекватно відображають поведінку

мікро- і нанорозмірних елементів на різних рівнях проектування наносистем. При виборі проектного рішення необхідно також враховувати вплив технологічного процесу і різних дестабілізуючих факторів. У зв'язку з цим для реалізації складних маршрутів проектування MEMC і NEMC застосовуються інтегровані САПР, що дозволяють виконати моделювання та оцінку різних проектних рішень на різних рівнях проектування. Методологія проектування MEMC базується на:

- класичній механіці, заснованій на трьох законах Ньютона і принципі відносності Галілея;
- теоретичній (аналітичній) механіці (механіка Лагранжа);
- класичній електродинаміки (рівняння Максвелла).

Методологія проектування NEMC базується на:

- квантовій механіці;
- наноелектромеханічних концепціях.

Проектування MEMC і NEMC майже на всіх своїх фазах автоматизовано. У зв'язку з цим розглядаються методології, алгоритми, методи опису і моделювання, які використовуються при автоматизованому проектуванні. Все перераховане вище об'єднується в поняття «CAE - computer aided engineering». Специфічні особливості проектування мікро- і наносистем в порівнянні з традиційними макросистемами впливають з їх розмірів.

9 ІНТЕГРОВАНІ ПАКЕТИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ MEMC І NEMC

Для автоматизованого проектування MEMC і NEMC необхідні інтегровані системи, що включають в себе підсистеми:

- проектування на компонентному рівні;
- проектування на міждисциплінарному рівні;
- проектування електронних компонентів;
- проектування з урахуванням забезпечення надійного функціонування MEMC і NEMC.

В даний час розроблені різні пакети автоматизованого проектування MEMC і NEMC, що дозволяють виконати проектування на всіх основних етапах в інтегрованому маршруті. Одним з таких пакетів є інтегрований пакет програм CoventorWare. В даний час він включає в себе наступні основні модулі (рис. 9.1):

- модуль ARCHITECT;
- модуль DESIGNER;
- модуль ANALYZER;
- модуль INTEGRATOR.

Є також додаткові модулі:

- модуль Etch-3D;
- модуль MEMULATOR.

Програма **ARCHITECT** містить бібліотеки поведінкових моделей електромеханічних, оптичних, рідинних та інших мікропристроїв і бібліотеки базових елементів (пружини, маси й демпфери, конденсатори, індуктивності, операційні підсилювачі, транзистори і т. д.).

Модуль моделювання дозволяє виконати різні види аналізу:

- розрахунок робочих точок по постійному струму;
- аналіз перехідних процесів;
- частотний аналіз в режимі малого сигналу;
- аналіз передавальних функцій по постійному струму;
- аналіз чутливості.

Програма **Architect** дозволяє значно полегшити працю проектувальника, надаючи різні засоби візуалізації MEMC і NEMC. При виборі тривимірного моделювання відкривається додаткове вікно CoventorWare - Architect 3D-Visualizer. Вікно складається з трьох панелей. На лівій панелі відображається дерево аналізу моделі. З

верхньої панелі конструктор вибирає необхідні інструменти для вибору проекції проектованого пристрою. У центральній панелі відображаються результати роботи. Наприклад, при моделюванні подвійного гіроскопа можна:

- збільшити зображення за допомогою лупи;
- вивчити поведінку проектованого пристрою в необхідний проміжок часу;
- отримати зображення проектованого пристрою в різних проекціях.

При цьому зміна кольору від синього до червоного відповідає зміні амплітуди зміщення з нульового до максимального значення. У програмі **Architect** доступна анімація для наступних видів аналізу: розрахунок сталих і перехідних режимів, а також модальний аналіз. Можливості анімації в програмі **Architect** значно посилилися після появи модуля Scene-3D. Продукт Scene-3D є допоміжною програмою для модуля Architect і дозволяє відображати схему МЕМС пристрою у вигляді тривимірного прототипу.

З введенням до складу пакета CoventorWare модуля Scene-3D стало можливим формування тривимірного виду розроблюваного пристрою, а також візуалізація його функціонування безпосередньо з модуля **Architect**, минаючи етап формування пошарових масок. Подібна функціональність стала можливою завдяки введенню спеціальної бібліотеки тривимірних елементів, пов'язаної з бібліотеками поведінкових моделей. Крім того, крім МЕМС пристроїв, 3D-бібліотеки можуть містити моделі будь-яких інших елементів, наприклад, транзисторів. Інтегрований пакет автоматизованого проектування CoventorWare підтримує формат gdsii, dxf. Можливий експорт моделі в програмний комплекс Ansys. Іншим прикладом інтегрованого пакета автоматизованого проектування МЕМС і НЕМС є САПР MEMS Pro. Він призначений для проектування таких МЕМС:

- механізмів (радіальні та лінійні гребні, зубчасті передачі, мембрани, консолі, мікровузли);
- інерційних блоків (якоря, балки, пластини);
- електромеханічних структур (гребневі мікродвигуни);
- п'єзореzystорів і п'єзоелектричних елементів і т. д.

В даний час інтегрований пакет автоматизованого проектування EMS Pro включає в себе наступні основні модулі:

- MEMS Modeler;
- L-Edit;
- Design Rule Checker.

Програма MEMS Modeler дозволяє отримати поведінкові моделі. Інтегроване середовище дозволяє виконати проектування на всіх основних етапах автоматизованого проектування MEMC і HEMC. Інтегрований пакет автоматизованого проектування MEMSPro підтримує формат gdsii, cif. Можливий обмін моделями з програмним комплексом Ansys.

Ще одним прикладом інтегрованого пакета автоматизованого проектування MEMC і HEMC є САПР IntelliSuite фірми IntelliSense Software, США. Підтримує маршрут проектування для наступних MEMC:

- мікропристрої (акселерометри, гіроскопи, мікрофони, запам'ятовуючі пристрої і т. д.);
- мікродвигуни (радіальні та лінійні гребневі мікродвигуни, зубчасті передачі, мотори і т. д.);
- електромагнітні MEMC (перемикачі, перебудовуються конденсатори і т. д.);
- НВЧ MEMC (компланарні хвилеводи, електрооптичні модулятори, мікросмужкові пристрої і т. д.);
- аналізатори (теплові, механічні, електростатичні, п'єзоелектричні, біоMEMC і т. Д.).

В даний час інтегрований пакет автоматизованого проектування IntelliSuite включає в себе наступні основні модулі:

- MEMaterial;
- IntelliFAB;
- AnisE;
- RECIPE;
- IntelliMask.

Інтегрований пакет автоматизованого проектування IntelliSuite підтримує формат iges, gdsii, dxf. Можливий обмін звичайно-елементними моделями з програмним комплексом Ansys.

10 ЗАГАЛЬНІ ВІДОМОСТІ ПРО КОМП'ЮТЕРНЕ МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ. КЛАСИФІКАЦІЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ

Основою автоматизованого проектування MEMC і НЕМC є математична модель. Вона являє собою безліч математичних об'єктів і відносин між ними, адекватно відбиває конкретні властивості MEMC і НЕМC.

Модель - матеріальний об'єкт, система математичних залежностей або програма, що імітує структуру або функціонування досліджуваного об'єкта.

Моделювання - представлення різних характеристик поведінки фізичної або абстрактної системи за допомогою іншої системи.

Математичне моделювання - метод дослідження процесів і явищ на їх математичних моделях.

Вивчення комп'ютерного математичного моделювання відкриває широкі можливості для усвідомлення зв'язку інформатики з математикою і іншими науками - природничими та соціальними.

Комп'ютерне математичне моделювання в різних своїх проявах використовує практично весь апарат сучасної математики.

До класифікації математичних моделей можна підходити по-різному, поклавши в основу різні принципи:

а) класифікація моделей по галузях наук (математичні моделі у фізиці, біології, соціології і т. д.);

б) класифікація моделей за вживаним математичним апаратом (моделі, засновані на застосуванні звичайних диференціальних рівнянь, диференціальних рівнянь в приватних похідних, стохастичних методів, дискретних алгебраїчних перетворень і т. д.);

в) класифікація моделей з точки зору цілей моделювання:

- 1) описові (дескриптивні);
- 2) оптимізаційні;
- 3) багатокритеріальні;
- 4) імітаційні.

Приклад. Змінюючи тепловий режим відпалу, можемо прагнути підібрати такий, щоб досягти максимальної провідності півки, тобто оптимізуємо процес.

Імітаційна модель - опис системи та її поведінки, що може бути реалізовано та досліджено в ході операцій на комп'ютері.

Імітаційне моделювання - дослідження поведінки складної системи на її моделі. Можна сказати, що найчастіше імітаційне моделювання застосовується для того, щоб описати властивості великої системи за умови, що поведінка складових її об'єктів дуже просто і чітко сформульовано. Математичний опис тоді зводиться до рівня статистичної обробки результатів моделювання при знаходженні макроскопічних характеристик системи. Такий комп'ютерний експеримент фактично претендує на відтворення натурального. Імітаційне моделювання дозволяє здійснити перевірку гіпотез, досліджувати вплив різних чинників і параметрів.

11 ЕТАПИ, ЦІЛІ ТА ЗАСОБИ КОМП'ЮТЕРНОГО МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ

Розглянемо процес комп'ютерного математичного моделювання, що включає чисельний експеримент з моделлю (рис. 11.1).

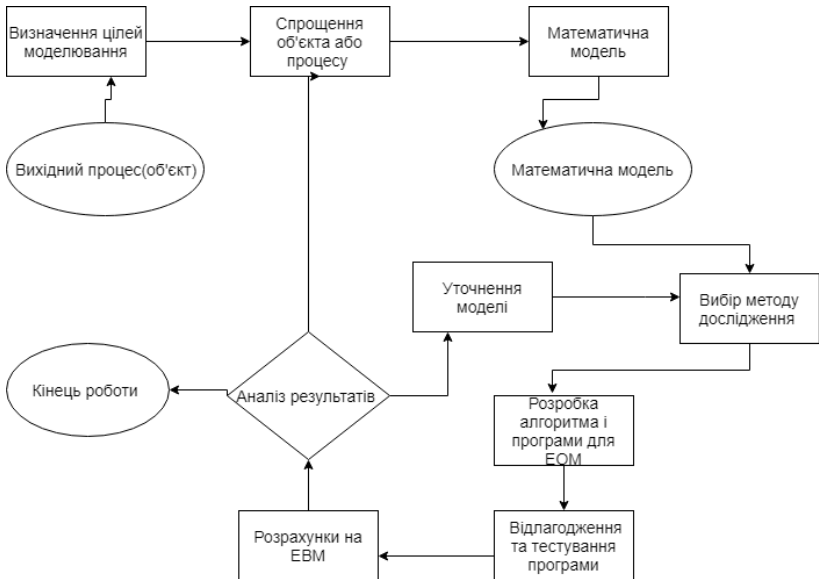


Рисунок 11.1 - Загальна схема процесу комп'ютерного математичного моделювання

Перший етап - визначення цілей моделювання.

Головні цілі:

а) модель потрібна для того, щоб зрозуміти, як влаштований конкретний об'єкт, яка його структура, основні властивості, закони розвитку і взаємодії з навколишнім світом (розуміння);

б) модель потрібна для того, щоб навчитися управляти об'єктом (або процесом) і визначити найкращі способи управління при заданих цілях і критеріях;

в) модель потрібна для того, щоб прогнозувати прямі і непрямі наслідки реалізації заданих способів і форм впливу на об'єкт (прогнозування).

Найважливішим етапом моделювання є розділення вхідних параметрів за ступенем важливості впливу їх змін на вихідні. Такий процес називається ранжируванням (поділом по рангах). Від того, наскільки вміло виділені найважливіші фактори, залежить успіх моделювання, швидкість і ефективність досягнення мети. Виділити більш важливі (або, як кажуть, значущі) фактори і відсіяти менш важливі може лише фахівець в тій предметній області, до якої належить модель.

Відкидання (принаймні при першому підході) менш значущих чинників робить об'єкт моделювання грубішим і сприяє розумінню його головних властивостей і закономірностей. Уміло проранжована модель повинна бути адекватна вихідного об'єкту або процесу щодо цілей моделювання. Зазвичай визначити, чи адекватна модель, можна тільки в процесі експериментів з нею, аналізу результатів.

Наступний етап - пошук математичного опису. На цьому етапі необхідно перейти від абстрактного формулювання моделі до формулювання, що має конкретне математичне наповнення. У цей момент модель постає перед нами у вигляді рівняння, системи рівнянь, системи нерівностей, диференціального рівняння або системи таких рівнянь і т. д.

Коли математична модель сформульована, вибирається метод її дослідження. Як правило, для вирішення однієї і тієї ж задачі є кілька конкретних методів, що розрізняються ефективністю, стійкістю та ін. Від вірного вибору методу часто залежить успіх всього процесу.

Розробка алгоритму та складання програми для ЕОМ – це творчий і складно формалізований процес. В даний час при комп'ютерному математичному моделюванні часто використовуються прийоми процедурно-орієнтованого (структурного) програмування.

При створенні імітаційної моделі можна також скористатися можливостями одного з пакетів математичної підтримки (Mathematica, MathCad, MathLab та ін.).

В даний час існують проблемно-орієнтовані імітаційні мови, в яких об'єднуються різні альтернативні підходи і які самою своєю структурою визначають можливу схему дій розробника моделі. Характерним прикладом такого роду є SLAM - simulating language for

alternative modeling - імітаційна мова для альтернативного моделювання. Після складання програми вирішуємо з її допомогою найпростішу тестову задачу (бажано, із заздалегідь відомою відповіддю) з метою усунення грубих помилок. Це тільки початок процедури тестування, яку важко описати формально вичерпним чином. По суті, тестування може тривати довго і закінчитися тоді, коли користувач за своїми професійними ознаками вважатиме програму вірною.

Потім іде власне чисельний експеримент, і з'ясовується, чи відповідає модель реального об'єкту (процесу). Модель адекватна реальному процесу, якщо деякі характеристики процесу, отримані на ЕОМ, збігаються з експериментальними із заданим ступенем точності. У разі невідповідності моделі реальному процесу повертаємося до одного з попередніх етапів.

12 ОСОБЛИВОСТІ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ ВИРОБНИЧИХ СИСТЕМ

Для аналізу виробничих систем, які дуже складні, різнопланові, не мають вичерпного математичного опису, адекватні математичні моделі, будь то логічні або числові, побудувати неможливо. Природним тут є використання методів імітаційного моделювання.

Система може бути однозначно описана набором значень виробничих параметрів, характерних для кожного конкретного її стану. Якщо їх внести в комп'ютер, то зміни в ході обчислювального процесу можна інтерпретувати як імітацію переходу системи з одного стану в інший. При таких припущеннях імітаційне моделювання можна розглядати як динамічне представлення системи шляхом просування її одного стану до іншого за характерними операційних правил.

При імітаційному моделюванні виробничих систем зміни їх стану відбуваються в дискретні моменти часу. Основна концепція імітаційного моделювання системи і в цьому випадку полягає в відображенні змін її стану з плином часу. Таким чином, тут визначальним є виділення і однозначне опис станів системи, що моделюється.

Імітаційні моделі дозволяють без використання будь-яких аналітичних або інших функціональних залежностей відображати складні об'єкти, що складаються з різноманітних елементів, між якими існують різноманітні зв'язки. У ці моделі також може бути включений і людина.

Без принципових ускладнень в такі моделі можуть бути включені як детерміновані, так і стохастичні потоки (матеріальні і інформаційні). За допомогою імітаційного моделювання можна відображати взаємозв'язки між робочими місцями, потоками матеріалів і виробів, транспортними засобами і персоналом.

Незважаючи на такі очевидні переваги, перш за все полягають в широті і універсальності застосування, при цьому методі з виду упускається існування логічних зв'язків, що виключає можливість повної оптимізації одержуваних на цій моделі рішень. Гарантується лише можливість відбору кращого з переглянутих варіантів.

Практично ж імітаційне моделювання в багатьох реальних випадках - єдино можливий спосіб дослідження. Після розробки імітаційної моделі з нею проводяться комп'ютерні експерименти, які дозволяють зробити висновки про поведінку виробничої системи.

Поява і розвиток методів комп'ютерного імітаційного моделювання стало можливим також і в результаті розвитку методу статистичних випробувань, що дозволив моделювати випадкові події і процеси, що займають велике місце в реальних виробництвах.

При складанні імітаційної моделі і проведенні з її допомогою моделювання досліджуваного об'єкта необхідне рішення кількох пов'язаних між собою завдань. До них відносяться:

- аналіз модельованої системи і складання її формалізованого опису, включаючи виявлення інформаційно-логічної структури системи, ідентифікацію її компонентів, вибір параметрів, що характеризують стан цих компонентів, розробку комп'ютерної моделі системи, здатної відтворити її поведінку, планування експерименту по розгортанню подій в комп'ютерній моделі, що відображають події в моделюється системі;

- розробка методології комп'ютерного статистичного експерименту, включаючи генерацію випадкових або псевдовипадкових чисел, імітацію різних випадкових подій, статистичну обробку даних;

- проведення власне комп'ютерного експерименту на імітаційній моделі, включаючи управління параметрами і змінними моделі в ході її дослідження на комп'ютері.

13 ОСНОВНІ ПАРАМЕТРИ, ЩО ВИЗНАЧАЮТЬ ПЕРЕБІГ І СТАН ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ВИРОБНИЦТВА КОМПОНЕНТІВ НАНО- ТА МІКРОСИСТЕМНОЇ ТЕХНІКИ

Сучасний аналіз всього технологічного процесу виробництва матеріалів електронної техніки здійснюють на основі системного підходу, де основним є поняття «велика система» або «велика технологічна система», що включає сукупність відбуваються фізико-хімічних і механічних процесів, об'єктів обробки і засобів для їх реалізації.

Технологічний процес, як велика система, схематично представлений на рис. 13.1, де:

$X_1, X_2, \dots, X_m$ - входи системи або параметри визначення вихідних продуктів;

$Y_1, Y_2, \dots, Y_n$ - виходи системи або вихідні параметри кінцевого продукту;

$Z_1, Z_2, \dots, Z_i$ - контрольовані та керуючі фактори, необхідні для підтримки певного технологічного процесу;

$W_1, W_2, \dots, W_k$ - неконтрольовані (збурюючі) фактори, що випадково вплив на рівновагу процесу.

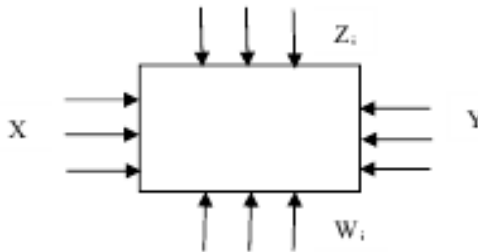


Рисунок 13.1 - Схема технологічного процесу як великої системи

Охарактеризуємо основну групу параметрів, що визначають перебіг процесу і його стан в будь-який момент часу.

1. Вхідні параметри.

До них відносяться величини, які можуть бути визначені експериментально, але можливість впливу на них протягом процесу відсутня. Значення цих установок не залежать від режиму процесу.

Наприклад, в процесах отримання монокристалів вхідним параметром може бути склад вихідної сировини, що завантажується в піч.

2. Вихідні параметри кінцевого продукту.

До них відносять ті характеристики системи, значення яких визначаються режимом досліджуваного процесу в системі, яка піддається впливу вхідних, керуючих і неконтрольованих чинників.

3. Керуючі параметри.

Це ті параметри процесу, на які можна впливати у відповідності з тими чи іншими вимогами, що дозволяє керувати процесом.

4. Неконтрольовані (збурючі) параметри.

Значення їх змінюються випадково з плином часу і зазвичай не можуть бути виміряні протягом процесу.

Формалізація системи здійснюється за допомогою математичної моделі, що виражає зв'язок між вихідними параметрами системи, параметрами стану і вхідними керуючими і впливаючими змінними. Вибір сукупності характеристик процесу визначається головним чином тими цілями, для досягнення яких і будується математична модель процесу.

14 СИСТЕМА МОДЕЛЕЙ В ТЕХНОЛОГІЇ НАНО- ТА МІКРОСИСТЕМНОЇ ТЕХНІКИ

Залежно від призначення виділяють нижченаведену систему моделей:

1. Моделі для дослідження процесів

Головною метою таких моделей є отримання повних знань про об'єкт в цілому або окремих його сторонах. Їх основу складає стисле, наочне і взаємопов'язане відображення накопичених уявлень про фізичні, фізико-хімічних та інших закономірності перебігу процесів. Особливістю моделей цього рівня є досить глибоке відображення окремих сторін і явищ (наприклад, аналіз гідродинаміки розплаву при вирощуванні кристалів). Основним призначенням таких моделей є отримання нових знань, проникнення у внутрішній механізм явищ і процесів. Другий важливий напрямок пов'язаний із можливістю використання цих моделей для синтезу алгоритмів керування і вибору ефективних керуючих впливів.

2. Моделі для розрахунку і оптимізації технології

Такі моделі застосовуються для розрахунків технологічних процесів і використовуються у вигляді інструкцій по веденню процесу в проєктованих і діючих агрегатах або в вигляді завдання установок автоматичним регуляторам. Для цієї мети можуть використовуватися або певним чином модифіковані моделі, описані вище, або створюватися спеціальні моделі в орієнтації на досягнення цільових технологічних критеріїв.

3. Моделі для прогнозування оптимальних траєкторій процесу в часі

В основу таких моделей можуть бути покладені уявлення, отримані за допомогою моделей двох розглянутих вище рівнів. Однак найбільш істотним тут є відображення особливостей протікання, наприклад, окремих процесів варіння скла, в залежності від конкретних початкових умов і стану з урахуванням взаємозв'язків керованих і внутрішніх параметрів, що може мати значний вплив на прогнозовані програмні режими. Ці моделі доцільно використовувати для «програвання» процесів в прискореному масштабі часу, наприклад, перед майбутнім варінням скла, коли стають відомими початкові умови (склад скла, температура і т. д.), для вибору оптимального розподілу керуючих впливів у часі (по ходу варіння).

4. Моделі для стабілізуючого або стежачого регулювання

Моделі цього рівня повинні відрізнятися найбільшою простотою й оперативністю відображення динаміки процесу. При побудові таких моделей широко використовується функціональний підхід. Вони найчастіше описуються залежностями лінійного виду і справедливі для управління по окремих каналах у відносно вузькому діапазоні зміни змінних.

5. Моделі для автоматизованих систем навчання і підвищення кваліфікації на основі тренажерних комплексів

Тренажери як засіб навчання мають такі основні переваги:

- заміна реального об'єкта навчальною інформаційною моделлю, що дозволяє виділити найбільш істотні для навчання сторони та елементи, в тому числі й аварійні ситуації, що рідко зустрічаються, вести підготовку поопераційно і поетапно з поступовим переходом від простого до складного;

- можливість зміни параметрів об'єкта, що моделюється і масштабу часу, що дозволяє підвищити інтенсивність навчання, наприклад, за рахунок багаторазового повторення необхідних ситуацій;

- різке скорочення матеріальних і енергетичних витрат на навчання в порівнянні з навчанням на реальному об'єкті та ін.

До моделей, призначених для тренажерних комплексів, в порівнянні з моделями для керування пред'являються менш жорсткі вимоги по кількісній адекватності, але в них необхідна більш висока ступінь якісної аналогії.

15 ПОНЯТТЯ ПРО ОБЧИСЛЮВАЛЬНИЙ ТЕХНОЛОГІЧНИЙ ЕКСПЕРИМЕНТ

Під **обчислювальним технологічним експериментом (ОТЕ)** розуміється така організація досліджень, при якій на основі математичної моделі проводиться вивчення технологічних пристроїв і процесів з допомогою ЕОМ, моделюється їхнє поведіння в різних умовах, знаходяться оптимальні технологічні параметри і режими діючих або проєктованих конструкцій. Ідеологічною базою ОТЕ є математичне моделювання, методологічною - теорія обчислювальних алгоритмів, технічною - сучасні електронні обчислювальні машини.

Необхідність використання ОТЕ викликана тим, що рішення сучасних науково-технологічних задач, що відрізняються надзвичайно складним математичним описом, традиційними методами стає скрутним, а в деяких випадках взагалі неможливим.

В даний час в програму проведення великих технологічних експериментів в обов'язковому порядку включається етап ОТЕ, що дозволяє виконати попереднє «прорахування» для уточнення і корекції плану проведення експерименту і прогнозування очікуваних результатів. Причому ОТЕ дозволяє провести дослідження в досить широкому діапазоні значень параметрів процесу без модифікації існуючих установок або розробки нових. Крім того, при комплексній взаємодії багатьох фізико-хімічних явищ і параметрів ОТЕ дозволяє розглядати вплив кожного явища або параметра на технологічний процес в окремо.

Розглянемо основні завдання ОТЕ, які вирішуються, наприклад, при проєктуванні технологічних процесів. Їх умовно ділять на наступні чотири групи:

- аналіз технологічних процесів;
- синтез технологічних процесів;
- діагностика;
- прогноз.

При аналізі технологічного процесу ОТЕ використовується для вирішення наведених нижче основних завдань:

1. Визначення «діапазону стійкого функціонування» технологічного процесу, тобто такої області значень керуючих параметрів, будь-які зміни яких в цій області не приводили б до порушення обмежень, накладених на вихідні параметри процесу.

2. Визначення критичних умов ведення технологічного процесу.

3. Встановлення ступеню впливу технологічних параметрів на режими ведення процесу і його вихідні параметри.

4. Дослідження впливу змін технологічних режимів процесу та конструкції технологічної установки на показники якості процесу.

5. Виявлення найважливіших точок технологічної установки, в яких необхідно розташування датчиків для контролю основних параметрів процесу і управління.

6. Дослідження характеру поверхні відгуку оптимізуються при проектуванні функцій для обґрунтування вибору методу пошуку екстремуму.

До групи завдань синтезу технологічних процесів можна віднести:

1) оптимізацію технологічних режимів процесу для діючої технологічної установки;

2) оптимізацію конструкційних параметрів діючої технологічної установки;

3) оптимізацію діючого технологічного процесу шляхом одночасної оптимізації технологічних режимів і конструкційних параметрів діючої технологічної установки;

4) проектування оптимального технологічного процесу.

З наведених завдань особливої важливості представляє остання, так як в цьому випадку застосування ОТЕ дозволяє вирішувати задачу значно більш загальну, ніж завдання оптимізації чинного технологічного процесу, що йде на вже існуючій установці.

При проектуванні нового процесу або вдосконаленні наявного без орієнтації на існуючу базову технологічну установку ОТЕ дозволяє проектувати оптимальний технологічний процес шляхом одночасного вирішення наступних завдань:

- проектування оптимальних технологічних режимів процесу;
- проектування оптимальних конструкційних і часових параметрів технологічної установки;
- проектування оптимальної системи управління процесом, включаючи систему датчиків.

До завдань діагностики відносяться:

1) визначення характеристик надійності технологічної установки і технологічного процесу в цілому;

2) діагностика правильності функціонування технологічного процесу;

3) дослідження стійкості функціонування технологічного процесу при наявності різного роду випадкових впливів на процес;

4) автоматизоване складання діагностичної таблиці.

До групи завдань прогнозу входять:

1) прогнозування показників якості технологічного процесу;

2) оцінка впливу передбачуваних технічних досягнень на показники якості технологічного процесу;

3) проектування і дослідження нових схем проведення процесу і нових технологічних режимів.

Наведений перелік завдань, звичайно, є узагальненим. При проектуванні конкретних технологічних процесів частина з перерахованих завдань може бути виключена або можуть бути введені нові завдання.

Для проектування технологічних процесів виключно важливе значення мають і супутні ОТЕ: по обробці даних натурного експерименту і по вивченню властивостей технологічного середовища.

Перший з цих ОТЕ реалізується в ході перевірки адекватності використовуваних в основному ОТЕ моделей з результатами натурних експериментів.

Другий супутній ОТЕ спрямований на отримання бази даних для основного ОТЕ, так як наявність інформації про параметри технологічного середовища дозволяє здійснювати пошук в досить широкому інтервалі значень параметрів процесу і оцінювати вплив змін, що вносяться до технологічного процесу, на показники якості процесу.

16 ЗАГАЛЬНА ХАРАКТЕРИСТИКА ТЕХНОЛОГІЇ СТВОРЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ. ЕТАПИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

До програмно-інструментальних засобів в першу чергу відносяться алгоритмічні мови і відповідні їм транслятори, потім системи керування базами даних (СКБД) з мовними засобами програмування в їх середовищі, електронні таблиці з відповідними засобами їх налаштування і т. д.

Розглянемо етапи розробки програм (рис. 16.1).



Рисунок 16.1 - Етапи розробки програмного забезпечення

Перший етап являє собою постановку задачі. На цьому етапі розкривається сутність завдання, тобто формулюється мета її вирішення; визначається взаємозв'язок з іншими завданнями; вказується періодичність рішення; встановлюються склад і форми подання вхідної, проміжної і результативної інформації.

Особливу увагу в процесі постановки завдання приділяється детальному опису вхідної, вихідної (результативної) і проміжної інформації. Особливість реалізації цього етапу технологічного процесу полягає в тому, що кінцевий користувач, що розробляється, добре знає її проблемну сторону, звичайно гірше уявляють специфіку і можливості використання ЕОМ для вирішення завдання. У свою чергу, предметна область користувача (особливо її окремі нюанси, здатні вплинути на рішення задачі) найчастіше незнайома розробнику програми, хоча він знає можливості і обмеження на застосування ЕОМ. Саме ці протиріччя є основною причиною виникнення помилок при реалізації даного етапу технологічного процесу розробки програм,

які потім неминуче відбиваються і на наступних етапах. Звідси вся важливість і відповідальність цього етапу, що вимагає здійснення коректної і повної постановки задачі, необхідності однозначного її розуміння як розробником програми, так і її користувачем.

Другий етап в технології розробки програм - математичний опис завдання і вибір методу її рішення. Наявність цього етапу обумовлюється рядом причин, одна з яких впливає з властивості неоднозначності природної мови, на якому описується постановка задачі. У зв'язку з цим на ньому виконується формалізований опис завдання, тобто встановлюються і формулюються засобами мови математики логіко-математичні залежності між вхідними та вихідними даними. Математичний опис задачі забезпечує її однозначне розуміння користувачем і розробником програми. Складність і відповідальність етапу математичного опису завдання і вибору (розробки) відповідного методу її рішення часто вимагають залучення кваліфікованих фахівців галузі прикладної математики, що володіють знанням таких дисциплін, як дослідження операцій, математична статистика, обчислювальна математика і т. д.

Третій етап технологічного процесу підготовки вирішення завдань ЕОМ являє собою алгоритмізацію її рішення, тобто розробку оригінального або адаптацію (уточнення і коригування) вже відомого алгоритму.

Алгоритмізація - це складний творчий процес. В його основу покладено фундаментальне поняття математики і програмування - алгоритм - кінцевий набір правил, однозначно розкривають зміст і послідовність виконання операцій для систематичного вирішення певного класу задач за кінцеве число кроків.

Будь-який алгоритм має наступні важливі властивості: детермінованість, масовість, результатність і дискретність.

Детермінованість алгоритму (визначеність, однозначність) - властивість, що визначає однозначність результату роботи алгоритму при одних і тих самих вихідних даних. Це означає, що набір вказівок алгоритму повинен бути однозначним і точно зрозумілий будь-яким виконавцем.

Масовість алгоритму - властивість, що визначає придатність використання алгоритму для вирішення безлічі завдань даного класу. Воно передбачає можливість варіювання вихідних даних в певних межах. Властивість масовості алгоритму є визначальним фактором,

що забезпечує економічну ефективність вирішення завдань на ЕОМ, так як для задач, рішення яких здійснюється один раз, доцільність використання ЕОМ, як правило, диктується позаекономічними категоріями.

Результатність алгоритму - властивість, що означає, що для будь-яких допустимих вихідних даних він повинен через кінцеве число кроків (або ітерацій) завершити роботу.

Дискретність алгоритму - властивість, що означає можливість розбиття певного алгоритмічного процесу на окремі елементарні дії. Таким чином, алгоритм дає можливість чисто механічно вирішувати будь-яке завдання з деякого класу однотипних задач.

Складання (адаптація) програм (кодування) є завершальним етапом технологічного процесу розробки програмних засобів. Він передусє початку безпосередньо машинної реалізації алгоритму розв'язання задачі. Процес кодування полягає в перекладі опису алгоритму на один з доступних для ЕОМ мов програмування.

У процесі складання програми для ЕОМ конкретизуються тип і структура використовуваних даних, а послідовність дій, що реалізують алгоритм, відбивається за допомогою конкретного мови програмування. Етапи тестування і налагодження функціонально пов'язані між собою, хоча їх цілі дещо відрізняються один від одного.

Тестування являє собою сукупність дій, призначених для демонстрації правильності роботи програми в заданих діапазонах зміни зовнішніх умов і режимів експлуатації програми.

Мета тестування полягає в демонстрації відсутності (або виявленні) помилок в розроблених програмах на заздалегідь підготовленому наборі контрольних прикладів.

Процесу тестування супроводжує поняття «налагодження», що має на увазі сукупність дій, спрямованих на усунення помилок в програмах, починаючи з моменту виявлення фактів помилковою роботи програми і завершуючи усуненням причин їх виникнення.

За своїм характером (причини виникнення) помилки в програмах діляться на синтаксичні та логічні.

Синтаксичні помилки в програмі є некоректну запис окремих мовних конструкцій з точки зору правил їх подання для вибраної мови програмування. (Помилки виявляються автоматично.)

Далі перевіряється логіка роботи програми на вихідних даних. При цьому можливі наступні форми прояву логічних помилок:

- в якийсь момент програма не може продовжувати роботу (виникає програмне переривання, зазвичай супроводжується вказівкою місця в програмі, де воно відбулося);
- програма працює, але не видає всіх запланованих результатів і не виходить на зупинку (відбувається її «заиклення»);
- програма видає результати і завершує свою роботу, але вони повністю або частково не збігаються з контрольними.

Після виявлення логічних помилок і усунення причин їх виникнення в програму вносяться відповідні виправлення, і налагодження триває. Програма вважається налагодженою, якщо вона безпомилково виконується на досить представницькому наборі тестових даних, що забезпечують перевірку всіх її ділянок (гілок).

Процес тестування і налагодження програм має ітераційний характер і вважається одним з найбільш трудомістких етапів процесу розробки програм. За оцінками фахівців, він може становити від 30 до 50% в загальній структурі витрат часу на розробку проектів і залежить від обсягу і логічної складності розроблюваних програмних комплексів.

Для скорочення витрат на проведення тестування і налагодження в даний час широко застосовуються спеціальні програмні засоби тестування (наприклад, генератори тестових даних) і прийоми налагодження (наприклад, метод трасування програм, що дозволяє виявляти, чи всі гілки програми були задіяні під час вирішення завдання з заданими наборами вихідних даних).

Після завершення процесу тестування і налагодження програмні засоби разом із супровідною документацією передаються користувачеві для експлуатації. Основне призначення супровідної документації - забезпечити користувача необхідними інструктивними матеріалами по роботі з програмними засобами. Склад супровідної документації звичайно обговорюється замовником (користувачем) і розробником на етапі підготовки технічного завдання на програмний засіб. Як правило, це документи, що регламентують роботу користувача в процесі експлуатації програми, а також містять інформацію про програму, необхідну в разі виникнення потреби внесення змін і доповнень до неї.

В процесі впровадження та експлуатації прикладних програмних засобів можуть виявлятися різного роду помилки, які не виявлені розробником при тестуванні і налагодженні програмних засобів. Тому при реалізації досить складних і відповідальних програмних комплексів за погодженням користувача (замовника) з розробником етап експлуатації програмних засобів може бути розбитий на два під-етапи: експериментальний (дослідний) і промисловий.

Сенс експериментальної експлуатації полягає у впровадженні розроблених програмних засобів на об'єкті замовника з метою перевірки їх працездатності та зручності роботи користувачів при вирішенні реальних завдань протягом досить тривалого періоду часу (зазвичай не менше року). Тільки після завершення періоду експериментальної експлуатації та усунення виявлених при цьому помилок і врахування зауважень програмний засіб передається в промислову експлуатацію.

Для підвищення якості робіт, оперативності виправлення помилок, що виявляються в процесі експлуатації програмних засобів, також виконання різного роду модифікацій, в яких може виникнути необхідність в ході експлуатації, розробник може за домовленістю з користувачем здійснювати їх супровід.

17 СУЧАСНІ МЕТОДИ РОЗРОБКИ ПЗ: МЕТОД СПАДНОГО ПРОЕКТУВАННЯ, МОДУЛЬНЕ ПРОЕКТУВАННЯ, CASE-ТЕХНОЛОГІЇ

Метод спадного проектування (метод покрокової деталізації, метод ієрархічного проектування, top-down-підхід).

Суть методу полягає у визначенні специфікацій компонентів системи шляхом послідовного виділення в її складі окремих складових і їх поступової деталізації до рівня, що забезпечує однозначне розуміння того, що і як необхідно розробляти і реалізовувати. Цей метод є незамінним при розробці складних за характером і великих за обсягом програм, коли до їх розробки необхідно залучати велику кількість програмістів, які працюють паралельно. Він дозволяє концентрувати увагу розробників на найбільш відповідальних частинах програми, а також полегшує можливість постійного контролю за її працездатністю у міру розробки, налагодження і об'єднання окремих складових програм за рахунок організації безперервності цього процесу протягом всієї розробки.

Для прискорення розробки програмного комплексу часто замість деяких програм нижнього рівня, які перебувають в процесі розробки, можуть застосовуватися спеціальні «програми-заглушки». Вони потрібні тільки на ранніх стадіях розробки для того, щоб не стримувати загальний хід створення програмного комплексу. Суть «програми заглушки» полягає в тому, що при зверненні до неї відповідно до заданого набором вихідних тестових даних вона не формує, а вибирає результат «рішення» з заздалегідь підготовленого набору. Завдяки цьому забезпечується можливість імітувати роботу на ЕОМ реально створюваної програми, отже, здійснювати перевірку працездатності програм верхнього рівня ще до того, як будуть розроблені і налагоджені всі складові програми нижнього рівня.

Модульне проектування

Реалізація методу спадного проектування тісно пов'язана з іншим поняттям програмування - модульним проектуванням, - так як на практиці при декомпозиції складної програми виникає питання про розумні межі її дроблення на складові частини. Разом з тим поняття модульності можна зводити тільки до подання складних програмних комплексів у вигляді набору окремих функціональних блоків.

Модуль - це послідовність логічно взаємопов'язаних фрагментів завдання, оформлених як окрема частина програми. При цьому програмні модулі повинні мати наступні властивості:

- на модуль можна посилатися (тобто звертатися до нього) на ім'я, в тому числі і з інших модулів;
- по завершенні роботи модуль повинен повертати управління тому модулю, який його викликав;
- модуль повинен мати один вхід і вихід;
- модуль повинен мати невеликий розмір, що забезпечує його видимість.

При розробці складних програм виділяють головний керуючий модуль, підлеглі йому модулі, що забезпечують реалізацію окремих функцій управління, функціональну обробку (тобто безпосередню реалізацію основного призначення програмного комплексу), а також допоміжні модулі, що забезпечують сервісне обслуговування пакета (наприклад, збір і аналіз статистики роботи програми, обробка різного роду помилкових ситуацій, навчання і видача підказок і т. п.).

Модульний принцип розроблення програм має наступні переваги:

- велику програму можуть розробляти одночасно кілька виконавців, і це дозволяє скоротити терміни її розробки;
- з'являється можливість створювати і багаторазового використання в подальшому бібліотеки найбільш вживаних програм;
- спрощується процедура завантаження великих програм в оперативну пам'ять, коли потрібно її сегментація;
- виникає багато природних контрольних точок для спостереження за здійсненням ходу розробки програм, а в подальшому - для контролю за ходом виконання програм;
- забезпечується більш ефективне тестування програм, простіше здійснюються проектування і подальша налагодження.

Переваги модульного принципу побудови програм особливо наочно проявляються на етапі супроводу і модифікації програмних продуктів, дозволяючи значно скоротити витрати сил і засобів на реалізацію цього етапу.

Структурне програмування

Актуальна для початкового періоду розвитку і використання ЕОМ проблема розробки програм, що займають мінімум основний пам'яті і виконуються за найкоротший час, в подальшому в зв'язку з

різким падінням вартості апаратної частини ЕОМ, значним зростанням їх швидкодії та обсягів пам'яті змінилася необхідністю розробки і застосування принципово нових методів складання програм. Все це знайшло своє втілення в розробці принципу структурного програмування. Однією з цілей структурного програмування було прагнення полегшити розробку і налагодження програмних модулів, а головне - їх подальший супровід і модифікацію.

В даний час структурне програмування - це ціла дисципліна, яка об'єднує кілька взаємопов'язаних способів створення ясних, легких для розуміння програм. Ефективність застосування сучасних універсальних мов програмування багато в чому визначається зручністю написання з їх допомогою структурних програм.

CASE-технології.

За останнє десятиліття в області засобів автоматизації програмування сформувався новий напрям під загальною назвою CASE-технології. CASE-технологія являє собою сукупність засобів системного аналізу, проектування, розробки і супроводу складних програмних систем, підтримуваних комплексом взаємопов'язаних інструментальних засобів автоматизації всіх етапів розробки програм. Завдяки структурним методам CASE-технологія на стадіях аналізу і проектування забезпечує розробників широкими можливостями для різного роду моделювання, а централізоване зберігання всієї необхідної для проектування інформації і контроль за цілісністю даних гарантують узгодженість взаємодії всіх фахівців, зайнятих в розробці ПЗ.

З самого початку CASE-технології розвивалися з метою подолання цих обмежень шляхом автоматизації процесів аналізу і інтеграції підтримуючих засобів. Вони володіють можливостями, що вказані нижче.

1. Єдина графічна мова

CASE-технології забезпечують всіх учасників проекту, включаючи замовників, єдиною строгою, наочною та інтуїтивно зрозумілою графічною мовою, що дозволяє отримувати доступні для огляду компоненти з простою і ясною структурою. При цьому програми представляються двовимірними схемами (які простіше у використанні, ніж багатосторінкові описи), що дозволяють замовнику брати участь в процесі розробки, а розробникам - спілкуватися з

експертами предметної області, розділяти діяльність системних аналітиків, проектувальників і програмістів, полегшуючи їм захист проекту перед керівництвом, а також забезпечуючи легкість супроводу і внесення змін в систему.

2. Єдина БД проекту

Основа CASE-технології - використання бази даних проекту (репозиторію) для зберігання всієї інформації про проект, яка може поділятися між розробниками відповідно до їх правами доступу. Вміст сховища включає не тільки інформаційні об'єкти різних типів, але і відносини між їх компонентами, а також правила використання або обробки цих компонентів. Репозиторій може зберігати понад 100 типів об'єктів: структурні діаграми, визначення екранів і меню, проекти звітів, опису даних, логіка обробки, моделі даних, їх організації та обробки, вихідні коди, елементи даних і т. п.

3. Інтеграція засобів

На основі репозиторію здійснюється інтеграція CASE-засобів і поділ системної інформації між розробниками. При цьому можливості репозиторію забезпечують кілька рівнів інтеграції: загальний користувацький інтерфейс по всім засобам, передачу даних між засобами, інтеграцію етапів розробки через єдину систему уявлення фаз життєвого циклу, передачу даних і засобів між різними платформами.

4. Підтримка колективної розробки та управління проектом

CASE-технологія підтримує групову роботу над проектом, забезпечуючи можливість роботи в мережі, експорт-імпорт будь-яких фрагментів проекту для їх розвитку та / або модифікації, а також планування, контроль, керівництво і взаємодія, тобто функції, необхідні в процесі розробки і супроводу проектів. Функції також реалізуються на основі сховища. Зокрема, через репозиторій може здійснюватися контроль безпеки (обмеження і привілеї доступу), контроль версій і змін і ін.

5. Макетування

CASE-технологія дає можливість швидко будувати макети (прототипи) майбутньої системи, що дозволяє замовнику на ранніх етапах розробки оцінити, наскільки вона прийнятна для майбутніх користувачів і влаштовує його.

6. Генерація документації

Вся документація щодо проекту генерується автоматично на базі сховища (як правило, відповідно до вимог діючих стандартів). Безсумнівна перевага CASE-технології полягає в тому, що документація завжди відповідає поточному стану справ, оскільки будь-які зміни в проекті автоматично відбиваються в репозиторії (відомо, що при традиційних підходах до розробки ПО документація в кращому випадку запізнюється, а ряд модифікацій взагалі не знаходять в ній відображення).

7. Перевірка проекту

CASE-технологія забезпечує автоматичну верифікацію і контроль проекту на повноту і спроможність на ранніх етапах розробки, що впливає на успіх розробки в цілому: за статистичними даними аналізу п'яти великих проектів фірми TRW (США), помилки проектування і кодування складають відповідно 64 і 32% від загального числа помилок, а помилки проектування в 100 разів важче виявити на етапі супроводу ПЗ, ніж на етапі аналізу вимог.

8. Автоматична генерація об'єктного коду

Генерація програм в машинному коді здійснюється на основі сховища та дозволяє автоматично побудувати до 85-90% об'єктного коду або текстів на мовах високого рівня.

9. Супровід і реінжиніринг

Супровід системи в рамках CASE-технології характеризується супроводом проекту, а не програмних кодів. Засоби реінжинірингу і зворотного інжинірингу дозволяють створювати модель системи з її кодів та інтегрувати отримані моделі в проект, автоматично оновлювати документацію при зміні кодів і т. п.

ПЕРЕЛІК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Щербакова, Е. Н. Информационные технологии нано- и микросистемной техники: учебно-методическое пособие для студентов специальности 1-38 01 04 «Микро- и наносистемная техника» / Е. Н. Щербакова. – Минск: БНТУ, 2017. – 78 с.

1 Яворський Н.Б., Теслюк В.М., Литвинова Є.І. Комп'ютерні методи в інженерії мікроелектромеханічних систем: навчальний посібник. - Режим доступу: <http://cad.lp.edu.ua/project/b1.pdf>

2 М. Лобур, М. Мельник. Основи мікросистемних пристроїв: навчальний посібник. - Режим доступу: <http://cad.lp.edu.ua/project/b3.pdf>

3 Семенець В. В., Невлюдов І. Ш., Палагін В. А. Введення в мікросистемну техніку та нанотехнології: підручник для студ. вищ. навч. закл.: затв. МОНУ. – Х.: Компанія СМІТ, 2011. – 416 с.

4 Норенков И.П. Автоматизированное проектирование: учебник для студентов вузов. – 2-е изд., перераб. и доп. – М., 2000. – 188 с. – Информатика в техническом университете

5 Норенков И. П. Основы автоматизированного проектирования: учебник для студентов вузов. – 2-е изд., перераб. и доп. – Москва: Изд-во МГТУ им. Н. Э.Баумана, 2002. – 336 с. – Информатика в техническом университете

6 Советов Б. Я., Яковлев С. А. Моделирование систем: учебник для вузов / 3-е изд., перераб., доп. – Москва: Высш.шк., 2001. – 343 с.

7 В.Теслюк, А.Зелінський, В.Каркульовський, Я.Василюк. Розширене проектування мікросистемних пристроїв: Навчальний посібник. - Режим доступу: <http://cad.lp.edu.ua/project/m6.pdf>

8 В. Хаханов, І. Хаханова, Є. Литвинова, М. Лобур, Г. Хаханова. Схемотехнічне проектування і моделювання НВІС. Підручник. – Режим доступу: <http://cad.lp.edu.ua/project/m1.pdf>

9 Качан О. Я. Принципы формирования микро- и наносистем. – 2016. – 284 с.

10 П. Кособуцький, М. Лобур, С. Чумаченко. Статистичне проектування мікроелектромеханічних систем. Навчальний посібник. – Режим доступу: <http://cad.lp.edu.ua/project/m2.pdf>

11 Мироненко И. Г., Суходольский В. Ю., Холуянов К. К. Автоматизированное проектирование узлов и блоков РЭС средствами

современных САПР: учеб. пособие для вузов / под ред. И. Г. МIRONENKO. – Москва: Высшая школа, 2002. – 391 с.

12 Нано- и микросистемная техника. От исследований к разработкам: сборник статей / под ред. П.П. Мальцева. – М.: Техносфера, 2005. – 592 с.

13 І. Ш. Невлюдов, В. А. Палагін. Мікросистемна техніка та нанотехнології. – Режим доступу: <http://openarchive.nure.ua/bitstream/document/9555/1/MST.pdf>