

БАЗОВІ КОМПОНЕНТИ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

1.1 Архітектура СППР та суміжні питання

Архітектура СППР визначається характером взаємодії основних її складових — інтерфейсу користувача; бази та сховища даних, документів і правил; моделей і аналітичних інструментів; інфраструктури комунікацій і мереж, а також елементів цих частин. Ефективне поєднання всіх елементів СППР дає змогу уникнути ряду труднощів щодо побудови СППР і підвищити продуктивність комп'ютерної системи за рахунок: особливої інтеграції бази даних СППР з іншими внутрішніми і зовнішніми базами даних; скорочення тривалості очікування відповіді на запит користувача; ефективного використання великих математичних моделей; вдалішої координації діалогу з базою моделей та базою даних; поліпшення розуміння програмістами окремих аспектів системи; зниження витрат на створення та експлуатацію системи; мінімізування вартості підтримки та збільшення продуктивності користувачів, включаючи уникнення збоїв системи та інших проблем щодо продуктивності; зменшення інфраструктурних перешкод, які затримують розгортання нових додатків інформаційних систем і технологій, особливо СППР.

Серед факторів, які впливають на вибір конкретної архітектури СППР, можна виокремити такі: необхідність подальшого розвитку комп'ютерної системи та адаптації, зокрема, за рахунок включення в неї наявних програмних засобів; застосування еволюційного підходу до розвитку СППР. Засоби підтримки прийняття рішень мають бути так розподілені на мережах, щоб досягати до творців рішень і забезпечувати необхідний захист інформації.

Мережа є важливим елементом інфраструктури, що найбільше сприяє функціонуванню системи підтримки прийняття рішень.

У сучасних СППР широко застосовуються такі головні мережеві технології:

- *Інтернет (Internet)*, який уможливорює з'єднання окремих індивідів у планетарному масштабі;
- *Екстранет (Extranet)*, що забезпечує зв'язок окремих компаній між собою.
- *Інтранет (Intranet)*, який призначений для з'єднання індивідів усередині компаній.

Складовою частиною архітектури СППР є проект мережі. Питання захисту СППР тісно пов'язані з їх архітектурою і мережевими альтернативами. Ці три теми тісно переплітаються і є дуже важливими з погляду побудови ефективної системи підтримки прийняття рішень. Якщо СППР не орієнтована на автономний (не підключений до мережі) комп'ютер у захищеному офісному середовищі, де він знаходиться під пильним оком менеджера, який його використовує, то потрібно обов'язково проводити сумісне розроблення архітектури СППР, організації мережі і розв'язувати питання стосовно захисту інформації. Можна поєднати головні компоненти СППР — інтерфейс користувача, базу даних, моделі й аналітичні інструментальні засоби, а також мережеву структуру СППР у загальну архітектуру СППР (див. рис. 1.1).

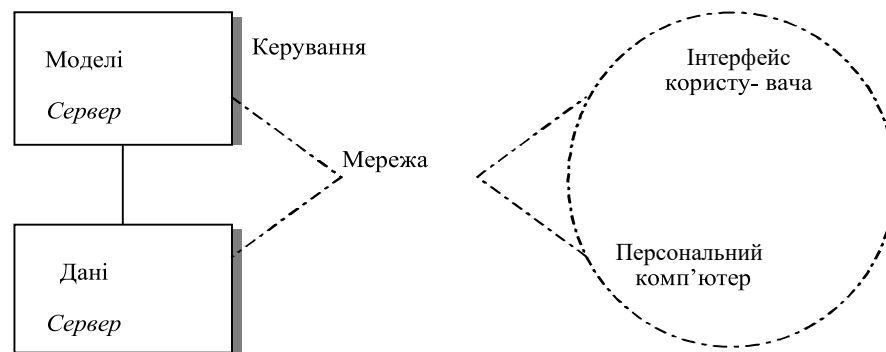


Рис. 1.1. Загальна архітектура СППР

Як буде показано далі в цьому розділі, головним компонентом у проекті СППР є інтерфейс користувача. До інструментальних засобів для побудови інтерфейсу користувача належать: симулятори інтерфейсу, СППР-генератори, інструментальні засоби запиту і звітів, пакет розроблення кінцевого користувача (front-end). Інтерфейси користувачів СППР можуть бути безпосередньо у клієнтів за архітектури «товстого клієнта» (thick-client) або доставлені мережею, використовуючи Web-сторінки чи Java applets (Java-додатки), в архітектурі «тонкого клієнта» (thin-client). Архітектура «тонкого клієнта», де користувач взаємодіє з використанням Web-сторінок, має багато переваг, але донедавна витонченість інтерфейсу користувача була обмежена принципами архітектури «товстого клієнта», коли програма зберігається в комп'ютері користувача СППР.

База даних СППР є сукупністю даних, які організовуються для легкого доступу до них і аналізу. Великі бази даних у СППР масштабу підприємств (корпоративних СППР) часто називають сховищами даних або вітринами даних. Документи або неструктуровані дані зберігаються інакше, ніж структуровані дані.

Web-сервери забезпечують потужну платформу для неструктурованих даних і документів. Архітектура для структурованої бази даних СППР в орієнтованих на дані СППР часто включає кілька серверів, спеціалізовані апаратні засоби і в деяких випадках програмне забезпечення як багатовимірних, так і реляційних баз даних. Розроблено багато ефективних методів виділення, перетворення, завантаження й індексації структурованих даних у СППР, а також є багато стратегій інжинірингу великих обсягів даних, якими є сховища даних.

Математичні моделі і аналітичні інструментальні засоби є важливою складовою багатьох СППР, особливо орієнтованих на моделі. Програмне забезпечення керування моделями може бути централізовано розміщеним з базою даних на сервері або специфічні моделі можуть розміщатися в комп'ютерах клієнтів. Додатки Java applets і програми JavaScript забезпечують могутні нові засоби доставлення моделей до користувачів в архітектурі

«тонкий клієнт».

Архітектура СППР і мережеві компоненти стосуються того, як апаратні засоби організовуються, як програмне забезпечення і дані розподіляються в системі і як компоненти СППР інтегровані й фізично з'єднані. Наявність чітко визначеної і добре комунікованої архітектури системи підтримки прийняття рішень забезпечує організацію значними перевагами. Вона допомагає співпраці розробників та сприяє вдосконаленню планування і виконання окремих кроків створення СППР.

Уся архітектура СППР має бути поданою у вигляді діаграм і бути зрозумілою перед тим, як прийматимуться конкретні рішення. Тип архітектури залежить від СППР. Маломасштабні

СППР, розроблені індивідами для їхнього власного використання, не потребують зусиль стосовно вищого архітектурного планування, хоча загальна архітектура інформаційної системи організації може впливати на можливості настільної СППР. Корпоративні (широкомасштабні) СППР вимагають ретельного планування архітектури для того, щоб вони мали успішне завершення. Слід зауважити, що поки що не створені єдині стандарти архітектури СППР, різні автори трактують це поняття по-своєму. Базовими типами архітектур систем підтримки прийняття рішень є: «мережа», «міст», «сандвіч», «башта».

1.2 Компоненти користувацького інтерфейсу

Призначення та загальні ознаки користувацького інтерфейсу

Важливість та ефективність користувацького інтерфейсу СППР

Комп'ютерні системи підтримки прийняття рішень призначені для розв'язування завдань користувачами, а тому невіддільною складовою їх роботи має бути точне дотримання вимог щодо деяких параметрів, здобутих від користувачів, урахування їх побажань за проектування системи. При цьому, якщо система функціонує коректно, але подає результати у спосіб, який є незручним для користувача, то роботу такої системи не можна вважати задовільною (людському фактору за створення СППР приділяється головна увага). Загальне побажання користувачів полягає в тому, щоб зі складними інформаційними системами можна було працювати успішно, обминаючи тривалий і дорогий етап навчання. Усе це зумовлює ряд вимог та особливостей щодо побудови користувацького інтерфейсу СППР.

Фактично для ОПР система підтримки прийняття рішень

— це інтерфейс користувача. Він охоплює всі механізми, якими команди, запити і дані вводяться в систему підтримки прийняття рішень, так само як і всі методи, якими результати чи будь-яка інша інформація виводяться системою. Якщо ОПР не має доступу до моделей і даних і не може легко переглядати результати, то система не може забезпечувати підтримку рішень. Тому, якщо інтерфейс не відповідатиме їхнім потребам і очікуванням, то ОПР часто повністю відмовлятимуться використовувати систему незалежно від потужності моделювання або придатності даних.

Ефективний інтерфейс користувача є важливим компонентом СППР будь-якого типу, але він особливо важливий для систем, які використовуватимуться безпосередньо менеджерами. У системах підтримки прийняття рішень інтерфейс користувача інколи називають діалоговим або «фронт-кінцевим» компонентом. Дехто міг би запитати: «чому інтерфейс користувача або компоненти діалогу СППР так важливі?» Дослідження показують, що коли інтерфейс СППР легкий у використуванні, то більшість людей розглядає систему як «дружню в користуванні» і тому більшою є ймовірність того, що менеджери дійсно використовуватимуть СППР.

Інтерфейс користувача — це фактично те, що менеджери бачать і використовують, коли вони взаємодіють з СППР. Можна дати конкретніше

визначення: інтерфейс користувача — це ряд меню, піктограм, команд, форматів графічного дисплея і/або інші презентації, які забезпечуються відповідною програмою, щоб дати змогу користувачеві мати зв'язок з СППР і використовувати її.

Інтерфейс користувача пов'язаний також з апаратними засобами і програмним забезпеченням, яке створює зв'язок і взаємодію між користувачем СППР і комп'ютерним процесором. Він містить відповідні різні графічні, звукові, сенсорні та інші засоби. Еволюція механізмів людино-машинної взаємодії показана в табл. 1.1.

Таблиця 1.1

ЕВОЛЮЦІЯ РОЗВИТКУ МЕХАНІЗМІВ ЛЮДИНО-МАШИННОЇ ВЗАЄМОДІЇ

Покоління ЕОМ	Характерні особливості користувацького інтерфейсу
Перше (1948—1955 роки)	Складний формальний стиль взаємодії; людина мала пристосовуватися до машини
Друге (1956—1963 роки)	Виявлення уваги до ергономіки користувацького інтерфейсу; створення деяких засобів графічного діалогу (імітаторів); застосування природної мови для виведення інформації
Третє (1964—1971 роки)	Поява примітивних (клавійатурних) засобів діалогу природною мовою, стандартних або формальних інтерактивних механізмів; зростання доступності графічних засобів
Четверте (1972—1979 роки)	Перехід до епохи орієнтації на користувача. База даних. Персональні комп'ютери як «слуги» чи «партнери» користувачів
П'яте (1980—1987 роки)	Ера експертних систем. Недорогі графічні інтерфейси. Інтегровані системи формального діалогу; досить розвинені системи з природною мовою; простота в користуванні, «дружність до користувача»
Шосте (після 1988 р.)	Злиття штучного інтелекту і людино-машинного діалогу, очікування появи «розумних» комп'ютерів, здатних розпізнавати ситуації інтуїтивним способом, робити індуктивні висновки та навчатися

Ефективний інтерфейс користувача є важливим, тому що дані і графічні

відображення на комп'ютерному екрані робочої станції надають контекст для людської взаємодії та забезпечують можливості для бажаних дій користувача. Користувач формулює відповідь згідно з контекстом розв'язуваного завдання і починає діяти. Дані повертаються назад до комп'ютера через інтерфейс. Добре розроблений інтерфейс користувача може збільшити швидкість оброблення інформації людиною, зменшити кількість помилок, підвищити продуктивність праці й створити у користувача відчуття повного володіння ситуацією. Якість інтерфейсу системи залежить від того, що користувач бачить або зчитує, що йому необхідно знати, щоб зрозуміти зміст зчитаної інформації, та які дії він має виконати в деяких випадках, щоб одержати потрібні результати.

Ключ до ефективного проекту інтерфейсу користувача полягає в поданні інформації так, щоб користувачі могли самостійно оцінити повний потенціал системи. Сьогодні це більше мистецтво ніж наука. З набуттям досвіду проектувальники стануть гармонійніше підходити до того, що потрібно користувачам, і зможуть краще задовольняти їхні потреби завдяки вдалій комбінації кольорів, правильному розміщенню вікон введення та виведення інформації і взагалі приємній робочій атмосфері.

Основні теми та механізми створення користувацького інтерфейсу

Щоб добре спроектувати інтерфейс, професіоналам з ІСМ потрібно ретельно вивчати запити потенційних користувачів. Обом групам потрібно бути поінформованими про важливі питання і теми, пов'язані з побудовою й оцінюванням інтерфейсу користувача. Ці теми стосуються: стилю користувацького інтерфейсу; проекту екрана і його компоновки; послідовності взаємодії людини з програмним забезпеченням; використання кольорів, ліній і графіки; інтенсивності потоку інформації; використання піктограм і символів; альтернативних пристроїв введення і виведення інформації. Менеджери й аналітики мають зосередити увагу на цій сімці питань щодо проектування СППР у разі, коли вони оцінюють прототипи або пропоновані екрани для СППР. Систематичне оцінювання інтерфейсу користувача СППР може по суті звузити межі невикористання СППР і сприяти розширенню масштабів її застосування.

Можна вирізнити три головні механізми для організації взаємодії користувачів із СППР:

- *формальний діалог*, що ґрунтується на «кмітливості» комп'ютера з урахуванням його структури як віртуальної машини;

- *природна мова*, яка відбиває особливості мислення конкретної людини, у результаті чого мова реалізується на лінгвістичній основі подання знань, комунікації і логічного висновку;
- *графічний діалог*, що відтворює задану предметну галузь, зокрема, із застосуванням піктограм (графічних зображень об'єктів чи дій). Останні мають деякий сенс для користувача, а в комп'ютері — це просто розміщення маркера.

Комп'ютер «надає перевагу» модальності формального діалогу, причому найперші обчислювальні системи розроблялися виключно для експертів з обчислювальної техніки; такі системи не мали ознак «дружності до користувача» і вимагали від серйозних користувачів знань однієї чи кількох мов програмування. Проте людям зручніше працювати з простішими в користуванні формальними засобами мовного інтерфейсу (меню, заповнення форм тощо), а також із графічними інтерфейсами і звичайною мовою.

Формальна і природна мови, графіка, а також різні комбінації цих елементів являють собою альтернативи для створення користувацьких інтерфейсів. До окремих інтерфейсних механізмів і критеріїв належать вибір із меню, кульковий маніпулятор (trackballs), керування за допомогою команд голосом, підказування голосом, клавішні керування курсором, клавіші фіксованих функцій, скролінг (вертикальні чи горизонтальні переміщення зображень у вікні екрана), керування вікнами на екрані дисплея, використання динамічних (електронних) таблиць, миша, визначені користувачем функціональні клавіші, сенсорні екрани та ін.

Важливою темою за створення сучасних користувацьких інтерфейсів є їхня адаптованість. В основу ідеї побудови *адаптивних інтерфейсів* покладено концепцію створення адаптивних програмних засобів, які можуть пристосовуватися до умов функціонування, непередбачених на етапі розроблення систем.

Стосовно створення інтерфейсу користувача СППР слід зазначити: кожна людина, яка працює з комп'ютером, має улюблені синоніми команд, свою інформацію, пристосовану до власних потреб, ураховуючи свій рівень знань, інтересів і навіть самопочуття в конкретний період доби. Ці фактори враховуються за побудови користувацьких інтерфейсів інформаційних систем, тобто ОПР надається можливість вносити до системи зміни, зумовлені особистісними сприйняттями інформаційного середовища. Незважаючи на те, що існують певні складності й технічні труднощі зі створенням і застосуванням адаптивних інтерфейсів (вони не можуть бути придатними до всіх ситуацій), перспектива їх впровадження в СППР може вважатися реальною.

Загальні висновки щодо користувацького інтерфейсу

Висновок про те, якому методу чи стилю користувацького інтерфейсу слід віддати перевагу, чи існує взагалі найпріоритетніший варіант інтерфейсу, значною мірою залежить від контексту рішень, предметних галузей, користувачів і завдань, що належать до цього процесу.

В інтерфейсах, які працюють під керуванням меню, можливі цілі СППР мають бути цілком передбачені розробниками системи, тобто бути вмонтованими чи заданими завчасно. У системах інтерфейсу на формальній мові користувачеві надається більше можливостей для досягнення цілей, поставлених перед СППР, але це досягається за рахунок вивчення кількох командних мов на рівні операційної системи чи на рівні робочих програм. Природна мова нині пробиває собі дорогу як потенційний життєздатний, робастий і складний інтерфейс для задоволення найвимогливіших потреб користувача.

Цінність графіки й кольору безпосередньо пов'язана з тим, наскільки ці елементи підтримують розв'язання завдань у СППР, зокрема, як вони впливають на якість та час прийняття рішення, використання інформації і на сприйняття користувача. Неодноразово було доведено, що режим подання дійсно впливає на ефект сприйняття та усвідомлення цінності інформації, але за чітко окреслених умов. Використання графіки зумовлює скорочення часу, необхідного для розв'язування задачі лише в тому разі, коли засоби одержання графічного звіту безпосередньо сприяють розв'язанню поставленої задачі. Багатокольорові протоколи створюють певні переваги, але також за певних умов.

Інтерфейс «користувач—система» забезпечує зв'язок ОПР із СППР та її компонентами. У свою чергу, користувацький інтерфейс має свої взаємопов'язані три компоненти або ключові аспекти:

мову дій — те, що може робити користувач під час спілкування із СППР. Мова дій охоплює операції від звичайного користування клавіатурою чи функціональними клавішами та сенсорними панелями до джойстика і усних команд звичайною мовою;

мову відображення — те, що бачить користувач у результаті роботи системи. Варіанти вибору щодо мови відображення досить різноманітні: використання різних принтерів, екранів, графічних засобів, кольору, графо побудовачів, звукового виводу тощо;

базу знань — те, що необхідно знати користувачеві, щоб вести діалог із

системою. Базу знань користувач може знати. Вона може бути надрукованою на папері (як посібник) або бути доступною як сукупність діалогових команд підказування (із застосуванням навчальних засобів) чи у вигляді деякої комбінації перелічених компонентів.

Питання про те, який конкретний метод чи пристрій користувацького інтерфейсу необхідний для мов дій і відображень у СППР, може розв'язуватися з двох поглядів — з погляду принципів і керуючих вказівок з проектування інтерфейсів інтерактивних інформаційних систем і з погляду врахування потреб потенційних користувачів.

Компоненти мови дій користувача

Мова дій (активностей) визначає форму введення, яку використовує ОПР, щоб ввести запити в систему підтримки прийняття рішень. Вона включає способи, якими користувачі отримують інформацію, подають запити про нові дані, вибирають моделі, задають чутливість і навіть отримують поштові повідомлення. Нині використовуються вісім головних типів мов дій, короткі характеристики яких наведено в табл. 1.2.

ТАБЛИЦЯ 1.2
ТИПИ ІНТЕРАКТИВНИХ ДІАЛОГІВ ЛЮДИНИ З КОМП'ЮТЕРОМ У СППР

Тип діалогу	Описання діалогу	Коментар
Запитання і відповіді	Комп'ютер ставить ряд запитань, на які відповідає людина	Ініціюється комп'ютером. Для абсолютно невідповідної особи — найменш здатний на помилки тип діалогу. З набуттям людиною досвіду стає обтяжливим
Заповнення форм	<i>Комп'ютер подає форми у вигляді бланків.</i> Людина заповнює бланки	Ініціюється комп'ютером. Швидший за діалог № 1, тому що людина дає кілька відповідей за одну транзакцію. Діалог найкращий, якщо в інформації, яку вводить людина, переважна більшість значень параметрів, а не команд. При частому користуванні діалог вимагає термінала з табуляцією

Закінчення табл. 1.2

Тип діалогу	Описання діалогу	Коментар
Вибір із меню	Комп'ютер дає список альтернатив. Людина вибирає одну чи кілька з них	Ініціюється комп'ютером. Діалог може застосовуватись для побудови команд і пошуку інформації в БД. Якщо є допустимий час для реакції системи і використовується вказівний пристрій вибору (наприклад, сенсорний екран), то діалог є цілком «природним»
Функціональні клавіші і командна мова	Людина здійснює бажану дію натискуванням клавіш, кожна з яких команда, модифікатор команди чи значення параметра	Ініціюється людиною, а також комп'ютером при використанні «програмованої» клавіатури чи навчаючих зображень. Придатний для невіддодотовленої особи, якщо синтаксис простий і (або) використовується тільки комп'ютерна ініціація; інакше діалог потребує підготовки людини
Ініційована людиною командна мова	Людина набирає команди, можливо, застосовуючи мнемонічну аббревіатуру	Діалог орієнтований на добре підготовлену особу, цілком володіючу моделлю, системними функціями і синтаксисом мови, наприклад, на розробника чи системного програміста
Мова запитів	Людина вводить запитання чи виконує процедури доступу до БД. Система формулює відповідь чи формує звіт	Діалог може використовуватися як новачками, так і програмістами; при цьому виникає багато помилок. Поки що немає детальних вказівок стосовно проектування діалогу мовою запитів
Природна мова	Діалог проводиться на звичайною для людини мовою	Ініціюється людиною або комп'ютером. Може мати місце змішана ініціація. Існують досить потужні засоби роботи природною мовою. Вартість діалогу досить висока, тому що система повинна мати великий обсяг знань

		з прикладної галузі для розуміння інформації людиною
Інтерактивна графіка	Генерування графічних зображень. Людина вибирає відображені об'єкти шляхом вказівок чи іншими немовними засобами	У разі швидкої реакції можливий досить ефективний і природний діалог, що виправдовує додаткові витрати (інтерактивна графіка відносно дорога) навіть у порівняно простих задачах (наприклад, пошук в ієрархічних БД)

Компоненти мови відображень (презентацій)

У той час як мова дій описує, як і якими засобами користувач повідомляє комп'ютер про необхідне оброблення, другий аспект користувацького інтерфейсу — мова відображень (показу, презентацій) описує, в який спосіб комп'ютер забезпечує отримання інформації користувачем. Звичайно, такий інтерфейс має передавати аналіз у такий спосіб, який є оптимальним для користувача. Це стосується не тільки результатів аналізу, але також і проміжних результатів на всіх стадіях прийняття рішення. Крім того, інтерфейс має забезпечувати відчуття контролю з боку людини щодо відтворюваного процесу і отримуваних результатів. Усе це має виконуватися приємним і зрозумілим способом.

Робота з вікнами (кадрування)

Спосіб організації інформації залежить від вигляду й типу моделей, ОПР і оточення, в якому ця особа працює. Основний принцип мови подання — вигляд форми показу інформації має бути «очищеним» і легким для сприйняття. Нині використання стандартів щодо проектування вікон для багатьох програмних виробів забезпечує їх приємним виглядом. Зокрема, цей стандарт відображає аналогію робочого стола, що складається з картотек. На екрані ми бачимо вікна, кожне з яких являє собою різний вигляд виведення даних. Одне вікно, наприклад, може містити діаграму, друге — електронну таблицю, а третє — текст допомоги користувачеві. На рис. 1.2 зображений приклад багатовіконного екрана. Використання кількох вікон для окремих видів інформації виділяє різні аспекти остаточного рішення.

Проектувальники мусять, однак, утримуватися від розміщення надмірної кількості вікон відразу на екрані, оскільки він стає загроможденим, дуже багато речей стають втраченими, і стає важче розглядати перспективу розв'язання проблеми загалом. Замість цього, якщо застосування дає змогу, то проектувальник має використати піктограми, щоб показати різні опції, як це ілюструється на рис. 1.3. Коли користувачі хочуть дослідити певний аспект проблеми, то вони можуть просто клацнути на відповідній піктограмі, щоб розкрити його для детального розгляду. Користувачам мають надаватися повні, але не завалені «робочі столи», метушня не має бути властивою застосуванню СППР.

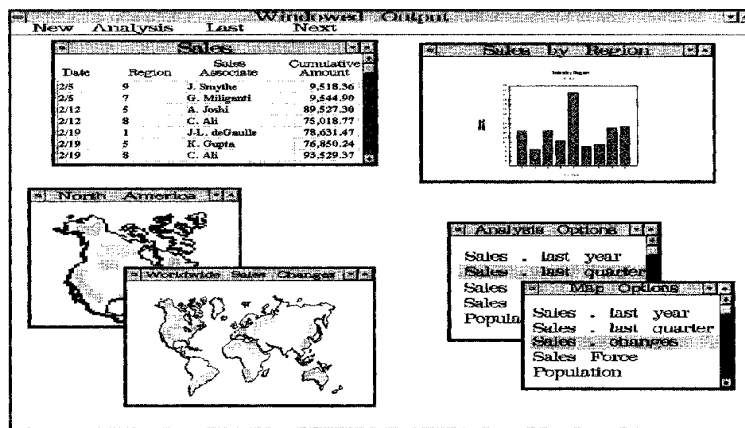


Рис. 1.2. Приклад режиму багатовіконного подання прийнятого рішення

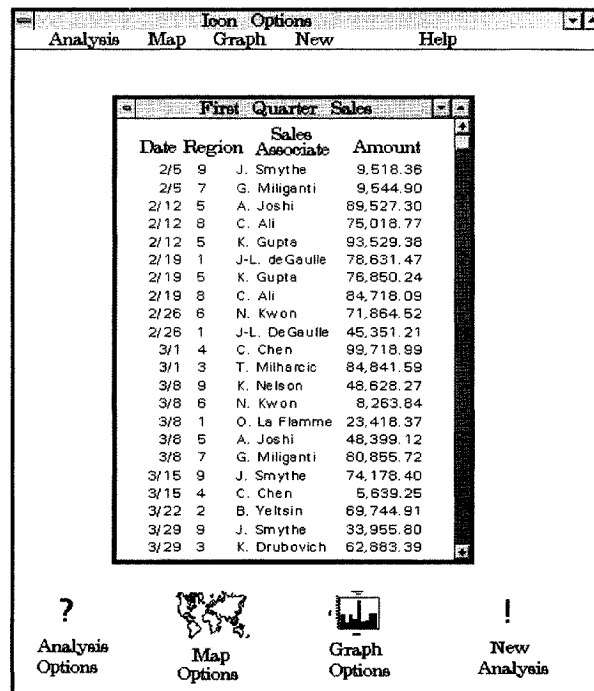


Рис. 1.3. Піктограмні опції

Зображення (образи)

Найзагальніше завдання виведення даних — показати результати деякого аналізу. Нехай, наприклад, мета полягає в тому, щоб показати збут у різних регіонах минулого року. Відповідна форма виведення даних залежить від того, що творець рішення сподівається робити з інформацією. Якщо ОПР просто хочуть знати, як різні зони збуту відповідають їхнім сподіванням, то вони могли б оцінювати результати, використовуючи зображення різних мордочок (metri-glyphs), як наприклад, показано на рис. 1.4. Ті з «обличчя, що усміхаються», показують сприятливий збут, у той час, як «сумні обличчя» відображають протилежне. Разом із цим, чим більша усмішка, тим більше збут перевищив сподівання, і, навпаки, чим сумніша гримаса, тим гірші результати.



Рис. 1.4. Мордочки (Metri-glyphs)

Можна навіть пов'язати одну множину результатів з «усмішкою», іншу — з «очима» обличчя. Наприклад, якщо усмішка відображає рівень прибутку, то очі могли б означати рівень дивідендів. Закриті очі асоціювалися б з невиплатою дивідендів, у той час, як ступінь відкритості очей міг би означати їх величину. Звичайно, не всі творці рішень (або не всі культури) розуміють переваги використання metri-glyphs, як одного із способів виведення даних.

Якщо мета аналізу полягала у визначенні місць, де збут був найбільший, то ми могли б показати це на карті різними штрихами або кольорами, як позначками місцевостей з відповідними результатами. Якщо метою аналізу було визначення динаміки збуту за окремі роки, то найбільш відповідним форматом виведення даних було б традиційне графічне подання результатів. Графік дає змогу точніше побачити, наскільки деякі зони збільшили збут, у той час як інші зменшили, а також визначити відносні оцінки (як наприклад «значно більше» або «незначно»).

З іншого боку, якщо ОПР хотіла б мати фактичні обсяги для обчислення збуту протягом деякого періоду, то графічне зображення не зовсім відповідає цьому завданню через те, що важко зібрати з різних джерел дані про збут. У такому разі таблиці надходжень є кориснішою формою виведення результатів.

Звичайно, інколи відповіднішою формою виведення даних могла б бути анімація (мультиплікація) і/або відео, ніж статичне відображення на екрані. Наприклад, якщо завдання полягає у моделюванні банку і в моделі змінюється кількість клерків, типів оброблених кожним з них послуг і кількість черг, то з метою виявлення впливу кожного з цих факторів на довжину черги анімація черг могла б бути ілюстративнішою, ніж наведення статистичних даних.

Персональне монопольне володіння аналізом

На додаток до надання можливості виводити відповідного типу результатні дані під контролем ОПР проєктувальники мають надати користувачам право персонального монопольного володіння аналізом, тобто відчуття того, що вони контролюють процес здійснення аналізу і, отже, їх вибір є повноважним. Комп'ютерні новачки не можуть відчувати власної участі у формуванні відповіді через те, що в них складається враження нібито це було «зроблено комп'ютером», а не ними.

Один із шляхів протидії цій тенденції — забезпечити користувачів легким засобом змінювати вид аналізу, якщо результати не відповідають запиту деякою мірою або цілком. Це досягається за рахунок розміщення на екрані кнопок, які відповідають різним варіантам аналізу. Наприклад, обчислення рівня рентабельності продукції можна виконувати двома шляхами: урахуваючи чи не враховуючи дисконт-фактор. Отже, користувач сам має вибирати форму аналізу, натискаючи одну з двох відповідних кнопок вікна. Або коли середня величина деякого показника може бути отримана автоматично, користувач може захотіти протестувати чутливість моделі сам, а не отримуючи відповідні результати автоматично, «всліпу». Тому має бути клавіша, яка давала б змогу переглядати доречні статистики на різних горизонтах часу.

Діаграми і зміщення

Оскільки необхідною умовою є об'єктивне використання моделей користувачами, то також важливо забезпечити їх неупередженим виведенням даних. Зміст і спосіб виведення інформації проєктувальниками може вплинути на те, як вона сприймається творцем рішення. Звичайно, проєктувальники не будуть навмисно спотворювати систему презентації, щоб забезпечувати тим самим упереджені результати. Однак, небезпечніша проблема виникає тоді, коли

спотворення з'являється ненавмисно.

Нехай, наприклад, користувач аналізує управління двома об'єктами, досліджуючи середню щоденну їх продуктивність. Якщо система виведення надає лише середні значення продуктивності, то при цьому може виникнути упереджене сприйняття виведення даних через те, що середні значення не допомагають користувачеві повною мірою бачити значимість показників. Скажімо, середня продуктивність у першому об'єкті може бути 6000 одиниць, а в другому — 8000. Ця різниця на перший погляд є суттєвою. Проте якщо одночасно система покаже, що середнє квадратичне відхилення щоденної продуктивності дорівнює 2000, то різниця не буде здаватися такою великою.

Інша проблема виникає тоді, коли проєктувальники із-за неувважності допускають зміщення в результатах, що відображені на діаграмах. Так, частіше за все творці рішень дивляться на діаграми, щоб одержати швидке враження від значень даних. Але часто таке враження знаходиться під впливом виду графіка, зумовленого, зокрема, різницею в масштабуванні координат. Наприклад, за допомогою простого збільшення масштабу осі абсцис графік може складати враження значно вищого темпу зростання за той же період. Аналогічно збільшення масштабу осі ординат робить рівень зростання немов би набагато меншим (графік розтягується). Проєктувальник мусить надати допомогу користувачеві, пояснивши, що подібне неадекватне висвітлення відбувається завдяки неправильному вибору одиниць вимірювання системи координат. Це також стосується і використання стовпчикових діаграм.

Іншим фактором, який може призвести до ефекту зміщення для творців рішень, є відсутність агрегації показників за створення гістограм чи кругових діаграм. З другого боку, агреговані значення можуть часто давати змогу творцям рішень узагальнювати неправильні дані. Важко перерахувати всі спотворення і зміщення, які можуть мати місце в діаграмах, однак розуміння досліджуваної проблеми може допомогти уникнути ефектів зміщення результатів у проєкті СППР.

Роль знань у користувацькому інтерфейсі

База знань або низка знань, що належить до інтерфейсу користувача, містить всю інформацію про систему, котру користувач мусить знати, щоб використовувати її ефективно. Ці знання стосуються того, які є команди для виконання операцій системи, щоб включити одне, ввести інше, вибрати опції чи змінити їх. Ці команди подаються на розгляд користувачам різними шляхами.

Попередня підготовка для використання системи могла б бути індивідуальною чи груповою, включати тренінг з клавіатурою та концептуальну підготовку. На додаток до цього тренінгу, звичайно, є можливість використовувати деякі позаекранні підказки і екрани допомоги з додатковою інформацією.

Загальна мета створення користувацького інтерфейсу полягає в тому, щоб зробити систему як можна простішою, щоб уможливити користувачам найповніше використання програмного забезпечення. Такий засіб має забезпечувати як досвідчених, так і недосвідчених користувачів, щоб надавати потрібну допомогу, зокрема, щодо застосування специфічних методів і моделей. Користувачі, зазвичай, не є фахівцями в статистичному чи фінансовому моделюванні, математичному програмуванні тощо. Їм потрібна допомога у формуванні їхніх моделей і використанні їх у належний спосіб. Така допомога має міститися в системі.

Питання проектування користувацького інтерфейсу

Загальні вимоги до проекту користувацького інтерфейсу

Мета проектування інтерфейсу користувача — це розроблення схеми або макету екрана і інтерфейсу, щоб вони були зручними для користування й візуально привабливими. Розроблювачі інтерфейсу мають приділяти значну увагу його ергономії, ставлячи за мету забезпечити комфортну й ефективну взаємодію користувачів зі складною системою оброблення інформації, а також досягти повноти знань, які використовуються в цьому процесі. Зацікавлені користувачі системи підтримки прийняття рішень і проектувальники інформаційних систем мають брати активну участь у процесах проектування і оцінювання інтерфейсів користувача СППР.

Перелічимо основні принципи, що зумовлюють певний стандарт інтерфейсу користувача.

1. Щодо засобів відображення й керування — домагатися, щоб уся відображувана інформація була легко зрозумілою і користувач постійно контролював ситуацію; передбачити засоби, що допомагають користувачеві пересуватися по СППР.

2. Щодо діалогу між користувачем і системою — мінімізувати складність завдань введення даних користувачем та ймовірність помилок введення. Передбачити альтернативні методи введення. Ретельно визначити процедуру оброблення помилок.

3. Підтримувати сумісність відображуваної інформації та діалогу по всій СППР.

4. З метою повторного входу до системи (якщо користувач перервав роботу із СППР) необхідно передбачити засоби зберігання виконаної роботи і забезпечити «дружній» режим повторного входу.

5. Передбачити спеціалізовані й умонтовані засоби протоколювання (разом з бібліотекою стандартних протоколів, доступних для користувача), а також діалогове відображення протоколів, діалогові засоби підказування для полегшення розуміння протоколів.

6. Мати на увазі, що для керівників ключовим засобом інтерфейсу є графічне відображення, перетворення табличних даних на графіки й діаграми.

7. Необхідно забезпечити точні та ефективні процедури керування базою даних для завдань підтримки великих масивів даних (включаючи засоби введення й оновлення), а також створити засоби супроводження даних, куди належать форми для введення даних,

і забезпечити можливість реєстрації транзакцій з метою перевірки.

8. Для прийому зовнішніх даних у СППР необхідно вмонтувати засоби інформаційного зв'язку. Оскільки принципи застосування користувацького інтерфейсу мають, як правило, узагальнений характер, то за виконання практичних робіт зі створення СППР необхідно урахувати також і потреби потенційних користувачів. Типи користувачів, завдань і ситуацій, пов'язаних з прийняттям рішень, мають визначити специфічні особливості всього процесу розроблення користувацького інтерфейсу.

Для керівників вищої ланки управління конче потрібна і є доречнішою зовсім інша, ніж для керівників середньої ланки, техніка користувацького інтерфейсу; СППР, орієнтовані на швидкість реакції або на кризові ситуації, мають зовсім інші вимоги до інтерфейсу, ніж системи підтримки довгострокового планування; альтернативні контексти задач (наприклад, чи передбачається використання СППР для підтримки структурування задачі або для одержання прогнозів) також потрібно враховувати у процесі синтезу або добору формальних засобів та інструментів для компонування мов дій і відображень людино-машинного інтерфейсу.

Отже, підтримка прийняття рішень у контексті побудови людино-машинних інтерфейсів має характер, явно зорієнтований на людські якості (специфіку роботи) користувача. Слід також зазначити, що сама проблема розуміння ролі СППР з боку користувача є неоднозначною, зокрема, розрізняють

пасивне й активне розуміння цієї системи.

Пасивне розуміння СППР стосується критерію простоти (дружнього ставлення) в користуванні або механізму користування системою, тобто роботи термінала, процедур введення і виведення, синтаксису використовуваного діалогу. Активне розуміння передбачає жорсткіші стандарти оцінювання системи, зокрема, ця форма розуміння потребує ставлення до СППР з погляду спроможності надання допомоги в розв'язаннях проблем, а також визначає ті характеристики інтерфейсу, які дійсно придатним до вимірювання способом підвищують можливості керівників у прийнятті рішень.

Але при оцінюванні СППР важливо аналізувати інтеграцію аспекту керування з боку користувача з процесом прийняття рішень за допомогою системи. Тому з метою створення ефективних інтерфейсів користувача СППР потрібні удосконалення в напрямі обох форм розуміння. Більше того, СППР може бути дуже дружньою з погляду комфортності роботи користувача чи навіть фактичного використання системи, але не впливати на якість прийнятих рішень. Можна також уявити ситуацію, коли СППР підвищує ефективність процесу прийняття рішень (особливо в тому разі, коли процедури і правила роботи менеджерів без системи є дуже суб'єктивними або мають очевидні дефекти), але при цьому дістає негативну оцінку з боку користувачів.

ROMC — підхід до проектування користувачького інтерфейсу

Спрага (Sprague) і Карлсон (Carlson) 1982 року запропонували підхід до проектування СППР і особливо до інтерфейсу користувача під назвою **ROMC**. Їхній підхід має чотири орієнтовані на користувача аспекти: 1) **Representations**: зображення інформації, що передається до користувача; 2) **Operations**: операції з маніпулювання відображуваними даними; 3) **Memory aids**: допомога для пам'яті користувача; 4) **Control aids**: забезпечення допомоги користувачу щодо контролю СППР, тобто засобами керування.

Хоча ROMC спочатку розглядався як підхід стосовно виявлення необхідних можливостей СППР, він також може служити для створення проектів екрана і для побудови інтерфейсу користувача СППР. Чотири компоненти цього підходу можуть удосконалити проектування екрана і його компоновку (розміщення, набір інструментів), тому він може бути корисним інструментальним засобом для проектування користувачького інтерфейсу СППР.

Хоча на відміну від проектування користувачьких інтерфейсів на персональних комп'ютерах, де розроблені стандарти користувачьких інтерфейсів

(див. наприклад, *Проектирование пользовательского интерфейса. Стандарт фирмы IBM/ Под редакцией М. Дадашова, DBS LTD, 186 с.*), для проектування користувацьких інтерфейсів СППР ще не створені стандарти, проте у розпорядженні розроблювачів СППР є різні керівні вказівки з проектування інтерфейсів, існують сотні загальних принципів і докладних специфікацій. Наприклад, Сміт і Моз'єр склали аж 679 конкретних вказівок для розроблення програмного забезпечення користувацького інтерфейсу інформаційних систем у такому плані: 1) введення даних; 2) відображення даних; 3) послідовність керування; 4) керівництво користувача; 5) передавання даних; 6) захист даних. Розроблені й інші підходи достворення користувацьких інтерфейсів СППР. Керівні вказівки для проектування діалогу й інтерфейсу користувача.

1992 року Шнайдерман (Ben Shneiderman) розробив кілька засадних принципів проектування, що, на його переконання, найпридатніші для діалогових систем. Ці основні принципи інтерфейсного проекту отримані евристичним шляхом завдяки досвіду проектування і використання СППР.

Постарайтесь узгодити. Цей принцип є одним із тих, що найчастіше порушуються, але є найлегшим із тих, які можна виправити і/або уникнути їх порушень. Узгоджені послідовності дій мають дотримуватися всередині подібних ситуацій; ідентична термінологія повинна використовуватися в командних рядках, меню і екранах допомоги; узгоджені команди мають застосовуватися всюди. Особливі ситуації мають бути зрозумілими і обмеженими кількісно. Система мусить виглядати, функціонувати і бути відчутною однаково всюди.

Забезпечте повсюдні скорочення (короткі форми, аббревіатури) для користувачів. Як тільки зростає частота використання, то користувач бажає зменшити кількість і збільшити темпи взаємодій. Часто грамотні й досвідчені користувачі цінують скорочення, спеціальні клавіші, невидимі (приховані) команди і засоби макровизначення. Коротша тривалість відповіді і швидші параметри показу є привабливішими для постійних користувачів. Система має реагувати на потреби користувачів, що відрізняються.

Забезпечте зворотний зв'язок (відгук). Для кожної дії користувача має бути деяка система зворотного зв'язку. Для частих і незначних дій відповідь може бути помірною, тоді як для рідкісних та головних дій відповідь має бути суттєвішою.

Проектуючи діалог, створи його припинення. Послідовності дій мають бути організовані в групи, які мусять мати початок, середину і кінець. Зворотний зв'язок у разі завершення групи дій має задовольнити користувача досягнутим чи він може стати сигналом до ігнорування відхилень дійсного функціонування від планів

роботи, або бути показником того, як підготуватися до наступної групи дій. Потужності діалогу і команд мають відповідати можливостям користувачів їх сприймати. Потужність є мірою кількості роботи, яка завершується даною командою до СППР.

Забезпечте просте відновлення у разі допущення помилок. Проектуйте систему так, щоб користувач не зміг зробити серйозної помилки. Якщо помилка зроблена, то система має її виявити і дати просту пропозицію та зрозумілі дії для відповідного виправлення. Користувачеві немає необхідності повторно вводити повну команду, йому скоріше потрібно тільки виправити помилкову частину команди. Помилкові команди мають залишати стан системи без змін, або система сама має дати команду про відновлення нормального стану.

Надайте змогу щодо легкої реверсивності (протилежності напрямку або зворотності) дій. Дії користувача мають бути реверсивними, тобто здатними до зміни напрямку на протилежний. Ця властивість забезпечує меншу тривогу користувача, оскільки він знає, що помилки можна легко виправити; це також заохочує досліджувати незнайомі опції та параметри. Елементами зворотності можуть бути прості дії, введення даних або завершена група дій.

Підтримуйте контроль з боку користувача. Досвідчені користувачі хочуть відчувати, працюючи з СППР, що вона завжди реагує на їхні дії. Система із сюрпризними діями, яка має нудні послідовності введення даних, її нездатність або труднощі з необхідності одержувати інформацію і виконувати бажану дію — все це створює невпевненість і незадоволення з боку користувача.

Зменшуйте інформаційне навантаження. Інформаційне навантаження є тією мірою, за якої може використовуватися пам'ять користувача, щоб обробити інформацію, відображену у вікнах дисплея. Це є метою виконуваного завдання, ознайомленості ОПР з ним та проекту інтерфейсу безпосередньо. Обмеження щодо здатності людини обробляти інформацію в режимі оперативної пам'яті потребує, щоб покази залишалися простими і протягом достатнього для тренування періоду, який є необхідним для вивчення команд (стану) і визначення послідовності дій. Проектувальники можуть зменшити інформаційне навантаження за допомогою забезпечення скоріше графічного, ніж буквенно-цифрового показу, та інших форматів, що відповідають вимогам отримання невідкладної інформації користувачами, використання легких для розуміння слів, які забезпечують прості діалоги.

Фактори, що впливають на успішне проектування користувацького інтерфейсу

Згідно з Ларсоном (Larson) на успішне проектування інтерфейсу користувача впливають одинадцять факторів. Деякі з них подібні до розглянутих щойно пропозицій Шнайдермана. До цієї системи факторів належать: тривалість виконання або проходження завдання в СППР; різноманітність або експлуатаційна гнучкість системи; якість допомоги, яка забезпечується; адаптивність; уніфікованість (однорідність) команд і інтерфейсу. До людських факторів можна віднести: тривалість навчання для здатності користування СППР; легкість повторного виклику (згадування); помилки, що допускають кінцеві користувачі; концентрація, яка вимагається; втома від застосування системи; задоволення, яке користувач має, використовуючи систему. Ларсон детально описує кожен із цих факторів, що так чи інакше впливають на інтерфейс користувача СППР.

Симулятор користувацького інтерфейсу

Корисним засобом для дослідження проектування і розроблення прикладних систем може бути симулятор користувацького(людино-машинного) інтерфейсу MMIST, який являє собою універсальну інтерактивну програмну систему, розміщену в пам'яті машини V AX11/780, з апаратно незалежним графічним пакетом. Розробники СППР можуть використовувати цю систему для створення і демонстрування інтерактивних засобів відображення, меню та послідовності оброблення інформації на стадії проектування програмного забезпечення життєвого циклу СППР.

Головними цілями симулятора MMIST є реалізація трьох можливостей, до яких належать:

- робастий інструмент побудови відображень чи меню, який забезпечує досить детальне подання;
- дружній інтерфейс, що задовольняє навіть недосвідчених користувачів;
- потужні засоби симуляції (імітації) інтерфейсу, за допомогою якого користувачі можуть досліджувати й оцінювати єрархію меню і форматів відображення ще до початку кодування робочих програм.

Система MMIST і інші симулятори інтерфейсу, створювані у вигляді ще

складніших версій, цілком сумісні з новими стратегіями побудови прототипів СППР. Ідея полягає в тому, щоб

дати користувачам можливість побачити, дослідити і навіть реально використати в інтерактивному, хоча і симульованому, режимі запропоновані засоби інтерфейсу для застосувань до СППР; потім інтерфейс може підлягати модифікаціям, кількість і обсяг яких визначається інтенсивністю зворотного зв'язку з користувачами.

Заключні зауваження

Приклади трьох типів інтерфейсів (меню-орієнтовані, графічні та на природній мові) не охоплюють всього діапазону різноманіття інтерфейсів; існує також багато змішаних типів і різних поєднань. Тому розробнику СППР важливо зорієнтуватись у цій множині можливостей; ключовим аспектом у такому разі має бути важливість створення сумісного і дружнього інтерфейсу користувача, який би функціонував ефективно й продуктивно, забезпечуючи зв'язок користувача з системою в цілому, з базою даних і базою моделей СППР, зокрема.

Закінчуючи описання проблеми взаємодії користувач — СППР зауважимо, що дослідження і роботи зі створення користувацького інтерфейсу проводяться в багатьох країнах, і тому можна сподіватися на появу досконаліших механізмів взаємодії користувача із системою (наприклад, проводяться дослідження інтерфейсу віртуальної реальності, де користувач взаємодіє з генерованим комп'ютерним трьохвимірним (3-D) середовищем, які зі зрозумілих причин у даному розділі не розглянуті.

База даних і система керування базою даних у СППР

База даних у СППР

Еволюція використання даних у СППР

Головну роль у будь-яких інформаційних системах, включаючи СППР, відіграють, як було показано в розділі 2, інформація і дані. Зазвичай дані зберігалися у файлах залежно від їх застосування. Це означало, що постійно в них відбувалися зміни і тому файли, пов'язані з кожною конкретною прикладною задачею, розв'язання якої потребує цих даних, також необхідно було змінювати.

Наприклад, нехай СППР розроблена для полегшення процесу планування на підприємствах. Вхідними даними для цієї задачі є попит на кожний продукт, що виробляється. Використовуючи традиційний файловий підхід, кожному менеджеру підприємства необхідно надавати інформацію про виробництво кожного продукту адміністраторам з оброблення даних у СППР, який модернізує відповідні файли. Потім кожному менеджеру з продажу необхідно надсилати інформацію адміністратору з оброблення даних стосовно обсягів продажу цих продуктів, включаючи інформацію про брак попиту на них. Отже, дані вводяться за допомогою деякого механізму у відповідному форматі згідно з їх застосуванням. Один або більше файлів надсилаються адміністратору з оброблення даних, який трансформує дані у формат, придатний для використання в СППР, та модернізує ці файли.

Частота такої модернізації залежить від потреб у своєчасності надходження даних, необхідності збереження даних з метою використання за призначенням та інтенсивності бізнес-процесів. Однак зрозуміло, що такий процес трансформації файлів, особливо, якщо файли зберігаються в різних форматах (що, головні, і трапляється), є в кращому разі неефективним. Помилки введення даних важко ідентифікувати у всіх прикладних задачах, а також важко гарантувати, що всі користувачі мають доступ до тих самих даних. Коли потреби різних прикладних задач змінюються та створюються нові поля або знищуються старі, то тоді означені вище проблеми ще більше ускладнюються, тим більше, що оскільки дані зберігаються в багатьох місцях, то необхідно розмножувати і засоби зберігання цих даних.

Як тільки корпорації зрозуміли важливість даних як загального ресурсу, вони активізували процеси збирання та зберігання даних. Одним з найзначніших досягнень інформаційних технологій було створення *корпоративних баз даних*, що являють собою сукупність взаємопов'язаних даних. Концепція такої бази даних — сумісне зберігання пов'язаних даних у форматі, незалежному від конкретної СППР. Коли зберігання та використання даних незалежні, то і рішення стосовно зберігання даних стають незалежними від рішень щодо їх використання. Ті, хто займається зберіганням даних, можуть сконцентрувати свої зусилля на мінімізації витрат на їх зберігання. Якщо дані зберігаються тільки на рівні корпорації, то обсяг витрат значно зменшується.

Крім того, різноманітні СППР можуть використовувати ті самі бази даних по-різному. Отже, інформація, яка фізично міститься в різних засобах зберігання даних, може бути поєднана в одному масиві для передавання на екрани

користувачів з мінімальними витратами часу та машинних ресурсів. Коли потреби прикладних задач змінюються, то добавлення або переміщення якогось поля даних може бути виконано достатньо ефективно. Більше того, рішення можна легше скоординувати, тому що кожний користувач використовує ту саму модифіковану версію даних. У такий спосіб, наприклад, інформаційна система рівня підприємства може використовувати дезагреговані дані про товарні запаси для визначення доступності конкретних матеріалів тоді, коли вони потрібні, у той час як відділ корпоративного планування може використовувати агреговані дані про товарні запаси для визначення можливості ефективнішого розміщення замовлень, що зменшує частоту оброблення даних.

У більшості корпорацій нині дуже мало ведеться дебатів щодо вибору між традиційним обробленням файлів та використанням технології баз даних. Хоча перехід від технології оброблення файлів до технології використання баз даних є важким та дорогим на початку, але коли вже цей перехід здійснено, то така технологія забезпечує гнучкість та узгодженість даних і використовує мінімальний обсяг пам'яті для їх зберігання. Такі переваги сприяють ширшому застосуванню СППР. Однак з позиції користувача ця технологія має деякі недоліки. Якщо файл розробляється для однієї прикладної задачі, то це дає змогу користувачу краще контролювати дані та мати швидкий доступ до них. Як тільки зберігання даних може бути адаптоване до конкретної прикладної задачі, то це забезпечує їх оброблення дещо легшим, дешевшим та швидшим способом. Усі ці переваги спостерігаються лише до тих пір, поки не зміниться сама задача чи потреби користувача. Тоді користувачі мають створювати все спочатку та перебудовувати вхідні дані, а це потребує додаткових зусиль і коштів. Усе це з урахуванням легкості поєднання даних, збільшення кількості доступних полів, довшим терміном використання та меншими затратами на зберігання даних допомагає зрозуміти переваги технології баз даних.

Особливості бази даних у СППР

Загалом базу даних можна визначити як сукупність елементів, організованих згідно з певними правилами, які передбачають загальні принципи описання, зберігання і маніпулювання даними незалежно від прикладних програм. Зв'язок кінцевих користувачів (прикладних програм) з базою даних відбувається з допомогою СКБД. Остання являє собою систему програмного забезпечення, яка містить засоби оброблення даних спеціальними мовами баз даних і забезпечує

створення бази даних та її цілісність, підтримує її в актуальному стані, дає змогу маніпулювати даними і обробляти звернення до БД, які надходять від прикладних програм і (або) кінцевих користувачів за умов застосовуваної технології оброблення інформації. До складу мов бази даних, які використовуються для вивчення і звертання до даних, належить мова опису даних (МОД) і мова маніпулювання даними (ММД).

Мова опису даних призначена для формування структури бази даних. Описання даних окремої проблемної галузі може виконуватися на кількох рівнях абстрагування, причому на кожному рівні використовується своя МОД. Опис на будь-якому рівні називається *схемою*. Найчастіше використовується трирівнева система, яка складається з *концептуального, логічного і фізичного рівнів*. На концептуальному рівні описуються взаємозв'язки між системами даних, що відповідають реально існуючим залежностям між факторами та параметрами проблемного середовища. Структура даних на концептуальному рівні називається *концептуальною схемою*. На логічному рівні вибрані взаємозв'язки відбиваються в структурі записів бази даних. На фізичному рівні розв'язуються питання організації розміщення структури записів на фізичних носіях інформації.

Мова маніпулювання даними забезпечує доступ до даних і містить засоби для зберігання, пошуку, оновлення і стирання записів. Мови маніпулювання даними, які можуть використовуватися кінцевими користувачами в діалоговому режимі, часто називають мовами запитів.

Бази даних і СКБД використовуються в будь-яких комп'ютерних системах. Проте порівняно зі звичайними підходами до реалізації бази даних для розв'язування деяких задач до функцій та інструментів БД і СКБД у контексті системи підтримки прийняття рішень висувається ряд додаткових і специфічних вимог.

Для використання СППР необхідний доступ до інформації зі значно ширшого діапазону джерел, аніж це передбачено у звичайних інформаційних системах. Інформацію потрібно діставати від зовнішнього середовища і внутрішніх джерел; потреба в зовнішніх даних тим більша, чим вищий рівень керівництва, яке обслуговується вибраною СППР. Окрім того, звичайні, орієнтовані на бухгалтерський облік дані (характерні для систем оброблення даних і адміністративних інформаційних систем) необхідно доповнити нетрадиційними типами даних, зокрема й такими, які досі взагалі не бралися до уваги за комп'ютеризації. Сюди належать: текстова інформація, матеріал систем автоматизованого проектування виробів і технологій, автоматизованого

виробництва, а також інші джерела інформації, необхідні для прийняття рішень.

Заслугує також на увагу особливість процесу «здобування і захоплення» даних у СППР на відміну від загальнішого процесу збирання даних із джерел. Призначення СППР потребує, щоб процес здобування (і СКБД, яка керує цим процесом) був достатньо гнучким, аби швидко здійснювати доповнення та зміни згідно з непередбаченими запитами, які надходять від користувачів. Для виконання процесу «здобування і захоплення» даних у сучасних СППР широко застосовуються програмні (інтелектуальні) агенти (див. розділ 9.6), засоби дейтамайнінгу (які докладно описані в розділі 9.3), а також сховища даних (див. розділ 10.2).

У системах підтримки прийняття рішень передбачається засіб, за допомогою якого користувач може налагоджувати базу даних згідно зі своїми особистими вимогами. З огляду на це існують процедури й команди для гнучкого переструктурування схем і схемної підмножини СКБД. Зауважимо, що сучасні програмні засоби для керування даними і СКБД характеризуються достатньою гнучкістю і простотою використання в межах колективу користувачів. Проте згадані засоби не можна переупорядкувати і пристосувати до конкретного користувача чи до розв'язування конкретної задачі з бажаною гнучкістю і досить малими витратами.

Підсистема даних у СППР

Схема підсистеми даних у СППР

Будь-яка система підтримки прийняття рішень містить підсистему даних, яка складається з двох основних частин: бази даних і системи керування базою даних (СКБД). Притаманний технології СППР акцент на оброблення неструктурованих і слабоструктурованих задач зумовлює деякі специфічні вимоги до цих елементів комп'ютерної системи. Насамперед ідеться про необхідність виконувати значний обсяг операцій з переструктурування даних. Потрібно передбачити можливість завантаження і наступного оброблення даних із зовнішніх джерел; функціонування СКБД у середовищі СППР на відміну від звичайної технології оброблення інформації в управлінських інформаційних системах потребує ширшого ряду функцій. Це стосується також і бази даних.

Схема підсистеми даних у СППР зображена на рис. 5.5. Основою СППР є корпоративна база даних. Такі системи забезпечують користувачів даними про велику кількість угод та справ повсякденного життя корпорації. Внутрішні бази

даних містять інформацію щодо продажу та купівлі товарів, витрат корпорації, персоналу, планів і прогнозів на майбутнє та інших аспектів діяльності організації. Потім ці дані можуть служити основою моделей у СППР. Але важливо те, що сьогодні цих службових записів про корпорації недостатньо для підтримки прийняття рішень.



Рис. 1.5. Підсистема даних СППР

Види баз даних у СППР

Зовнішні дані. У нинішньому столітті ОПР не можуть приймати ефективні рішення за браку інформації про один або більше факторів, що знаходяться за межами корпорації та належать до *зовнішніх даних*. Такими зовнішніми даними можуть бути, наприклад: інформація про надання переваг покупцями щодо деяких товарів, попит на продукцію конкурентів у конкретних регіонах, дані перепису населення або статистичні звіти різних галузей.

Публічні (загальнодоступні) дані. Деякі дані збираються та зберігаються в самих корпораціях. Наприклад, відділ маркетингу може збирати інформацію щодо поділу ринку та/або демографічні дані стосовно власних споживачів. Ці дані можуть акумулюватися роками та використовуватися у разі розв'язання багатьох маркетингових завдань. Інші дані доступні з публічних джерел. Деякі бази даних доступні тільки за допомогою комутатора або пошуку шляхом прямого доступу через корпоративні мережі. Інші бази даних є доступнішими через Інтернет, інколи за визначену плату. З кожним днем у всьому світі спрощується доступ до баз даних,

що можна використовувати для підтримки прийняття рішень. Документи уряду, копії промов, торгові угоди, газети та інші джерела новин можна знайти та скопіювати в персональний комп'ютер. Навіть книжки, довідкові матеріали та журнали доступні в електронній формі, які використовуються як додаткова інформація в процесі прийняття рішень.

Існує багато шляхів допомоги ОПР щодо використання таких публічних баз даних. По-перше, якщо бази даних зберігаються в корпорації, то розробникам необхідно гарантувати, що доступ до них може забезпечуватися через СППР так само, як і до інших корпоративних даних. Аналогічно, якщо база даних знаходиться в Інтернеті, то розробникам необхідно забезпечити не тільки інтерфейс доступу до неї, але також і зв'язок з Інтернетом через мережу чи модем. На щастя, коли зв'язок з Інтернетом налагоджено, то існує багато різних засобів для полегшення добування даних. Наприклад, розробники, що використовують Lotus Notes (див. розділ 11.2.4), можуть застосовувати гнучкі засоби пошуку та оброблення інформації, яка знаходиться на Web-сторінці. Слід додати, що можливості мови доповнення гіпертексту версії 3.0 (HTML 3.0) та вище дають змогу розробникам вмонтовувати деякі функціональні програми у Web-сторінки.

Приватні дані. Не всі дані зберігаються в базах даних спільного користування. Більшість ОПР використовують власні практичні методи підтримки прийняття рішень, використовуючи дані, які вони збирають самостійно для одержання стратегічної переваги у своїй корпорації. Інколи вони просто зберігають записи політичних процесів у організації та інформацію стосовно того, як вони можуть вплинути на когось або хтось може вплинути на них за допомогою конкретного рішення. В дійсності ОПР за формулювання альтернативних рішень застосовують ці додаткові дані з метою полегшення власного процесу прийняття рішень.

За умови, що СППР розробляється для реального забезпечення підтримки прийняття рішень, вона має полегшити розвиток та використання приватних баз даних. Тобто система має допомагати ОПР створювати та заповнювати ці бази даних, а також забезпечувати легкий доступ до них з метою пошуку потрібних даних та здійснення необхідних обчислень. Якщо система знаходиться в комп'ютері, то забезпечити доступ до неї легко. Навіть якщо система міститься в мейнфреймі або в середовищі розподіленої системи, то існує можливість підтримувати приватні бази даних на окремих ЕОМ. Однак коли це все зроблено, то вирішальним питанням стає забезпечення достатнього захисту бази даних від несанкціонованого її використання, тобто потрібно гарантувати, що тільки ОПР

може мати доступ до цієї інформації.

Системи керування даними в СППР

Обговорення концепції СКБД у СППР

Як уже зазначалося, раніше комп'ютерні системи створювалися з використанням підходу, ґрунтованого на обробленні файлів. Коли організації почали рухатися в напрямі до більшої комп'ютеризації оброблення своїх даних та технологічних процесів і впровадження сучасніших технологій, то вони почали відмовлятися від здійснення операцій з окремими файлами на користь інтегрованих баз даних. Це означає, що сукупність взаємопов'язаних корпоративних даних консолідується та організується у такий спосіб, щоб бути доступною для різноманітних користувачів.

Для підтримки оброблення інформації з використанням корпоративних баз даних були створені системи керування базами даних (СКБД). СКБД служить буфером між потребами прикладних задач та фізичним зберіганням даних. Вона знаходить і вибирає дані з місця їх фізичного розташування та надає їх у розпорядження конкретної програми у спосіб, сформульований у запиті.

Головною перевагою, яку забезпечує СКБД, є **незалежність фактично існуючого розміщення** (як вони подані фізично) та **вигляду даних** (у якому вони надаються для розв'язування прикладної задачі). СКБД забезпечує передавання даних для прикладної задачі у такий спосіб, що програмісти, які її розробляють, можуть сприймати таку організацію даних як визначену. Коли модифікуються функціонуючі прикладні задачі або створюються нові, їх просто необхідно «прикріпити» до СКБД, що значно зберігає термін модернізації. Навіть процес доповнення бази даних новими полями значно легший, ніж доповнення новими полями традиційних файлів.

Розглядаючи технологію баз даних з позиції перспектив розвитку СППР, необхідно зазначити, що не всі структури баз даних є однаковими за гнучкістю та/або практичністю. Існують три фундаментальні класичні структури баз даних — *єрархічна, мережева (або сітьова) та реляційна*, — кожна з яких має свої переваги та недоліки. Ураховуючи всі їхні переваги та недоліки, можна висновувати, що реляційна структура є найперспективнішою з погляду дальшого розвитку СППР. Розроблені також *семантичні моделі даних*. Розглянемо коротко класичні структури моделей баз даних.

Моделі баз даних

Єрархічні (або деревоподібні) моделі баз даних забезпечують відносно ефективне подання даних у СППР. Вони базуються на принципі підпорядкованості і являють собою деревоподібну структуру, яка складається із *вузлів (сегментів)*, які розташовані на різних рівнях єрархії, і *дуг (гілок)*. Кожен вузол — це сукупність логічно взаємозв'язаних атрибутів, які описують певний об'єкт предметної галузі, а неорієнтовані дуги показують інформаційні зв'язки між об'єктами.

Єрархічна модель упорядкована згідно з правилами, за якими розташовуються сегменти і дуги моделей. До таких правил належать:

1. На найвищому рівні єрархії знаходиться один вузол — *кореневий*. Кожен екземпляр кореневого вузла породжує відповідний логічний запис, тому пошук даних в єрархічній базі даних здійснюється за принципом «зверху вниз» (зворотного напрямку пошуку в єрархічних моделях немає).

2. В єрархічних моделях підтримуються лише співвідношення між елементами даних типу «один до одного» (1 : 1) або «один до багатьох» (1 : Б).

3. Взаємозв'язки в єрархічних базах даних будуються за принципом «вихідний—породжений», тому доступ до кожного вузла (за виключенням кореневого) здійснюється через його вихідний екземпляр, у зв'язку з чим шлях доступу до кожного вузла є унікальним і лінійним за структурою. Кожен породжений вузол може мати лише один вихідний.

4. Кожен вузол може мати кілька екземплярів конкретних значень атрибутів. Кожен екземпляр породженого вузла зв'язаний з екземпляром вихідного. Кожен екземпляр кореневого сегменту разом з багатьма взаємозв'язаними екземплярами породжених сегментів утворює один *логічний запис*. Якщо в цьому ланцюжку відсутній хоча би один екземпляр, то подібний запис існувати не може, і потрібно розв'язати питання про введення якихось штучних екземплярів.

Крім цих загальних правил кожна єрархічна СКБД може мати свої особливості і вносити обмеження щодо побудови моделі бази даних; ці обмеження можуть стосуватися кількості рівнів єрархії, кількості атрибутів у сегменті тощо. Багато СКБД, основаних на єрархічних моделях, містять механізми для оброблення додаткових відношень.

У **сітьовій (мережевій)** моделі бази даних відношення між типами записів не обмежуються єрархією, а можуть утворювати граф з поіменованими дугами і

вершинами. Домінуючою сітьовою моделлю є модель, розроблена групою КОДАСИЛ; у ній відношення між типами подаються в термінах теорії множин.

У **реляційній моделі бази даних** базова структура даних подана у вигляді плоскої двохмірної поіменованої таблиці, яку називають

«відношенням». Реляційне «відношення» включає поіменовані стовпчики-атрибути і рядки, які називають кортежами (записами). Зв'язки між реляційними «відношеннями» мають не статичний, а динамічний характер і встановлюються саме на період розв'язання задачі. Тому ця структура даних, з одного боку, є гнучкішою, оскільки немає необхідності визначати зв'язки між відношеннями на схемі, а з другого боку, така гнучкість може виявитись неефективною через те, що структуру попередньо не визначають.

Кількість зв'язків між відношеннями не лімітована, єдиною умовою створення зв'язку є наявність у відношеннях, що зв'язуються між собою, спільних атрибутів — ключових або атрибутів зв'язку. Відношення мають бути подані в третій чи четвертій нормальній формі, тому попередньо потрібно виконати процедуру *нормалізації* відношень.

Нормалізація відношень являє собою ітераційний зворотний процес декомпозиції вихідного відношення на кілька простіших відношень меншої вимірності. В цьому процесі необхідно дотримуватись таких вимог: усі атрибути мають бути атомарними (неподільними); між атрибутами не повинні існувати неповно функціональні, транзитивні і багатозначні залежності; у базі даних мусять мати місце не надмірні дублювання атрибутів, що виконують ролі зв'язку між реляційними відношеннями.

З погляду застосування в СППР реляційна модель будується на базовій моделі індивідуальних записів. Вона дає змогу проводити операції над записами, зокрема, вводити нові записи, обновлювати поля, викреслювати наявні записи, а також утворювати і викреслювати відношення, зв'язувати чи об'єднувати два або більше відношень на основі спільних атрибутів. Можна вибирати записи за наявністю ознак певних відношень і проекції, що забезпечують вибір підмножини полів, які належать до відношення. Крім того, беззаперечною перевагою реляційних моделей баз даних є простота і гнучкість у проектуванні. Вони можуть підтримувати не лише дані, але і знання про певну предметну галузь. Вітчизняний ринок програмних продуктів пропонує кілька десятків реляційних СКБД, які можна застосовувати в СППР.

Існує досить широкий клас **семантичних моделей баз даних**, включаючи прямі розширення класичних моделей баз даних, математичних моделей і багатьох

інших. До найвідоміших представників цього класу належить семантична реляційна модель даних «об'єкт—зв'язок» (entity—relationship), яка уможлиблює графічне відображення об'єкта, і семантичні єрархічні моделі, що розширюють реляційні моделі, забезпечуючи оброблення таких семантичних понять, як «класифікація», «агрегація», «узагальнення» й «асоціація».

Аналіз СКБД для СППР

Існують десятки готових програмних систем для реалізації компонента СКБД системи підтримки прийняття рішень. Ураховуючи специфічні особливості та вимоги щодо цих програмних продуктів, а також значну їх вартість, можна висновувати, що вибір конкретної системи керування базами даних не є тривіальним для створеної СППР і заслуговує на саму серйозну увагу. Тим більше це важливо, оскільки продовжуються теоретичні та прикладні дослідження в цій галузі, а потенційному користувачеві доводиться мати справу з масою рекламних матеріалів. Тому існує проблема створення когерентної (зрозумілої) і комплексної основи оцінювання програмного забезпечення СКБД, як механізму для структурування процесу порівняння й вибору з різноманітних альтернативних варіантів СКБД за всім спектром апаратної бази. Найвідомішим способом розв'язання цієї проблеми є схема Захеді.

Схема Захеді має єрархічну структуру і включає п'ять рівнів порівняння: *цілі, аспекти, компоненти, засоби і примітиви*. Критерій першого рівня містить цілі користувача щодо СКБД і цілі, які мають бути реалізовані в СКБД, щоб остання була прийнятною для СППР. Кожна СКБД як виріб має чотири аспекти: функціональний, фізичний, вартість і корисність. Аспекти містять компоненти, які, у свою чергу, поділяються на засоби, а останні — на примітиви, тобто досить прості модульні задачі СКБД, кількість яких може досягати 116.

Схема Захеді розроблена на основі низхідної структури оцінювання, яка не залежить від прийнятої моделі бази даних, і містить два типи аналізу: порівняння елементів альтернативних СКБД на однакових рівнях і поєднання оцінок на різних рівнях відповідно до прийнятих асоціацій.

Бази моделей і системи керування базами моделей у СППР

Моделювання і його роль у підтримці прийняття рішень

Дані та моделі є центральними елементами СППР. Фактично СППР відрізняється від інформаційних систем менеджменту наявністю інтерактивних програм (з їх допомогою користувач може досліджувати і «мандрувати» по базах даних різних форм, розмірів і типів) та бази моделей (усередині її користувач може конструювати, аналізувати, інтерпретувати одну чи кілька моделей).

Протягом 60-х років ХХ ст. управлінські інформаційні системи розроблялися на базі процедурних елементів, причому керування моделями полягало в керуванні бібліотеками процедур розв'язків, поданих у вигляді програм і підпрограм. На початку 70-х років набула стабільності концепція баз даних, на основі якої і створювалися інформаційні системи. Проте тепер визнаною є думка, що саме моделі визначають відношення між даними і суто базовий підхід до проектування систем призводить до нехтування зв'язків, які відповідають процесам і процедурам у середовищі розв'язуваних задач. Тому спостерігається перехід від концепції баз даних до підходу, що ґрунтується на базі моделей, які стають джерелом нових тверджень і підґрунтям для усвідомлення суті інформаційних відношень. Концепція керування моделями усвідомлюється дедалі ширшим загалом дослідників і спеціалістів як передній край у галузі інформаційних систем і систем підтримки прийняття рішень.

Моделювання взагалі і комерційне моделювання зокрема (саме воно переважно використовується в СППР) — це спрощення деякого явища з метою розуміння його суті. Метою цього моделювання є спрощення вибору стосовно того, що особа, яка приймає рішення, має чітко розуміти свій вибір з усіма наслідками, що випливають з нього. Коли статистики будують регресійну модель, то їхньою метою є визначення головних, суттєвих факторів. Фахівці з маркетингу, наприклад, використовують регресію для того, щоб передбачити попит на певний вид продукту. Вони знають, що безліч чинників впливають на процес прийняття рішення покупцями: купувати чи не купувати даний продукт. Тому розвиток маркетингових досліджень є корисним для того, щоб знати, чи буде подобатися, наприклад, молодим фахівцям або службовцям їхній продукт, і чи буде попит на цю продукцію відрізнитися у різних регіонах країни.

Моделі являють собою важливу частину СППР. Багато ділових рішень зумовлені великою кількістю факторів, що впливають на процес їх прийняття. З

цього впливає, що творці рішень мають відділяти необхідні компоненти даної ситуації від тих, які не мають до неї відношення. До того часу, поки потреба у такому «фільтруванні» буде очевидною, ніхто не буде вважати таку модель зручною для використання. Дуже часто важко сказати, яка модель є найвідповіднішою для визначеного процесу прийняття рішень. Інколи очевидно, який вид моделі потрібний за певних умов, але даних для підтримки рішення недостатньо. Врешті, отриманий результат не є тим, який має бути в дійсності, але деякою мірою ця модель буде відображати визначену ринкову ситуацію.

Незважаючи на те, що модель може використовуватися і без СППР, все таки її можливості зростають з використанням СППР завдяки *гнучкості, зручному інтерфейсу і можливостям створення запитів*. Раніше склалося так, що ОПР мали довіряти іншим особам, які розробляли та інтерпретували для них моделі. Це пов'язано з різноманітністю виконання комп'ютерних програм, що потребують застосування відповідних моделей. Разом із СППР нині творці рішень отримують власний доступ до необхідних моделей і даних, а також безпосередній доступ до результатів.

Це є легкий і «дружній» спосіб, який робить моделі, побудовані в СППР, досить привабливими. ОПР мають розуміти свою причетність до власного вибору та змінювати свій вибір, коли він може бути несумісним із тими фактами, які наперед відомі. До цього можна додати те, що у зв'язку зі швидкістю та зручністю аналізу творці рішень можуть отримати більше альтернатив для того, щоб знайти вигідніше рішення. Більше того, модель заохочує користувачів СППР досліджувати змінні, які є чутливішими до допущень стосовно проблеми, що розв'язується.

5.1.1. База моделей у СППР

Будова бази моделей у СППР

СППР може містити різні типи моделей. Наприклад, до статистичних моделей належать: регресійний аналіз, аналіз змінних величин і експоненціальний розподіл. Бухгалтерські моделі включають цінові моделі, бюджет, податкові плани і вартісний аналіз. Моделлю з управління персоналом є, наприклад, розподіл обов'язків.

До маркетингових моделей належать рекламний стратегічний аналіз та модель вибору споживача. Характеристики всіх цих моделей відрізняються залежно від міри їх використання. Кожна з них являє собою певне спрощення

процесу прийняття рішення, яке є корисним для розуміння елементів суті даного явища. Для цього необхідне вміння будувати і використовувати ці моделі, і робити це так, щоб менш досвідчені користувачі могли ефективно їх використовувати. Частина вимог для створення СППР — це знання того, що необхідно включати в моделі і як вони мають бути доповнені, щоб зробити їх зрозумілішими та доступними для ОПР.

База моделей СППР містить оптимізаційні і неоптимізаційні моделі. До оптимізаційних моделей належать моделі математичного програмування — лінійного (розподіл ресурсів, оптимальне планування, аналіз сітьових графіків, транспортна задача), нелінійного, динамічного; моделі обліку; моделі аналізу цінних паперів для визначення інвестиційної стратегії; моделі маркетингу та ін.

До неоптимізаційних належать статистичні моделі (лінійний і нелінійний аналіз регресій); методи прогнозування (аналізу) часових рядів; альтернативні методи моделювання (наприклад, машинна імітація) тощо.

Узагалі вся множина моделей в СППР, котрі входять до бази моделей, може бути відображена в трьохвимірному просторі з такими вимірами: *подання, час, методології*, як зображено на рис.1.6 .

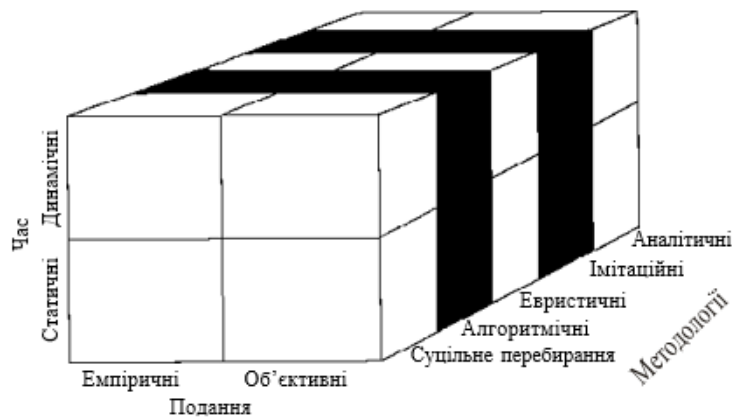


Рис. 1.6. Трьохвимірний простір класифікації моделей

Подання моделей

Перший вимір — «подання моделі» — описує тип даних, які необхідні для моделі і які зумовлюють необхідні підходи для збирання та оброблення даних. Відомо, що розрізняють моделі, які базуються на емпіричних і об'єктивних даних. Відмінність між ними полягає в процесі, яким генерується дана модель, а не у

відповіді, яка отримується.

Емпіричні моделі (в яких використовують дані досвіду) базуються на підготовці й поданні інформації людьми як індивідуально, так і в груповому порядку. Ці моделі можуть включати судові вироки, експертні думки та суб'єктивні оцінки. В експертному моделюванні інформація використовується у такий спосіб, яким створює рішення ОПР. Однією з серйозних проблем, що пов'язана з використанням таких моделей, є їхня суб'єктивність. Якщо дві особи спробують використати ту саму модель поведінки, то вони можуть досягнути різних результатів, тому що кожен з них керується різним досвідом і може по-різному його оцінювати.

На противагу експертним, *об'єктивні моделі* спираються на специфічні відокремлені дані і на їх аналіз за допомогою досконалої методики. Вони вважаються об'єктивними тому, що дані та спосіб, за допомогою якого вони використовуються, є специфічними, постійними та незалежними від досвіду прийняття рішень. Прикладом такої моделі є Reuters Money Network System (рейтерська грошова мережева система). Ця система дає змогу ОПР мати доступ до інформації про теперішній стан акцій, до попередніх даних і моделей для аналізу даних. Покриття інвестицій, розраховане одним користувачем для визначеної ситуації, буде таким самим, як і повернення інвестицій, розраховане іншим користувачем для іншої ситуації. Отже, тут немає суб'єктивних зв'язків за аналізу ситуації. Ми можемо впливати на результати рішення за допомогою вибору змінних, зміни періоду або груп шаблонів.

Як емпіричні, так і об'єктивні моделі мають певні переваги та недоліки. Перевагою об'єктивних моделей є те, що вони можуть вільно застосовуватися та доповнюватися новими даними. Крім того, для їх використання не вимагається великого досвіду. Об'єктивні моделі мають певні обмеження. Головною передумовою створення цих моделей є те, що для побудови математичної моделі спрощується реальна ситуація, хоча це і не виключає контроль необхідних факторів середовища, в якому приймаються рішення. Загалом, такі важливі фактори як конкуренція, регулювання цін і технології подаються у спрощеному вигляді. Якщо всі ці фактори з часом значно змінюються, то математична модель не буде доречною, тому що справжня суть середовища, в якому приймаються рішення, і його можливі реакції не будуть ураховані. За цих обставин надійніше використовувати емпіричні моделі.

Деякі СППР уможливають інтегрування об'єктивних і емпіричних моделей. СППР допомагає користувачам використовувати комбіновані моделі за

допомогою постійних спостережень та оцінювань напрямів розв'язувань і попереджень ОПР за умов появи небезпечних напрямів розв'язань.

Вимір часу

Вимір часу показує, як часто береться до уваги середовище, в якому приймається рішення. Тут важливу роль відіграють статичні та динамічні моделі.

Статична модель завчасно відображає картину того, як усі фактори впливають на середовище прийняття рішень. За такого моделювання вважається, що всі фактори залишатимуться тими самими, тобто в цих моделях допускають ситуацію, коли немає залежності від рішень, які будуть прийняті пізніше.

У динамічних моделях береться до уваги середовище прийняття рішень протягом визначеного періоду. В них можна розглядати однакові явища протягом різних періодів або взаємопов'язані рішення, які будуть прийняті в майбутньому.

Методологічний вимір

Третій вимір показує, у який спосіб дані (не має значення, об'єктивні чи емпіричні) будуть збиратися та оброблятися. Існує п'ять узагальнених методологій: суцільне перебирання (завершеного переліку); алгоритмізація; евристика; імітація (симуляція) та аналітична методологія.

За допомогою методології *суцільного перебирання* збирається та оцінюється найважливіша і найдорожча інформація про всі можливі альтернативи. За загальних обставин методологія завершеного переліку є непрактичною. Але існує кілька випадків, для яких ця методологія є необхідною та доречною. Так, перепис населення України є прикладом цієї методології. Згідно з нею всі жителі України ідентифіковані та пораховані. Метою перепису населення є відображення руху населення і завдяки цьому можна визначити щільність населення в країні. Щоб не досліджувати кожні райони в окремих областях, уряд прийняв рішення ідентифікувати кожного жителя індивідуально. Правда, за перепису населення не враховувалися безпритульні особи, тому, строго кажучи, цей перепис не був суцільним перерахунком.

Методологія завершеного переліку є також корисною для використання нейромереж файлів угоди для розпізнавання образу. Наприклад, нейромережева система була сконструйована для Malon-банку в Чикаго з метою ідентифікувати підозрілі кредитні картки, що могли бути краденими. За допомогою перевірки всіх

угод нейромережа ідентифікувала зміни в дрібних купівлях, як індикатор вкрадених кредитних карток. Це був метод завершеного переліку угод, доповнений шаблонно-розпізнавальними можливостями, який дав змогу системі швидко реагувати на появу в ній кримінальних явищ (докладно про нейромережіописано в розділі 9.4).

Другий підхід — алгоритмічна модель — являє собою розвиток процедур, які можуть повторюватися, і зрештою будуть

визначати бажану характеристику для прийняття рішень. Такі моделі найкраще подані в дослідженнях операцій. Алгоритми мають режим повторювальних обчислень (ітерацій), які можуть бути інструментом для знаходження кращого розв'язку. Ряд обчислень безпосередньо базується на особливостях певної проблеми. На противагу суцільному перебору варіантів алгоритм ідентифікує сприятливу інформацію, яка може бути використана для вибору найкращого результату без попереднього оцінювання всіх можливих розв'язків задачі.

Третій можливий вид моделювання — *евристика*. У загальному випадку, евристика застосовується для розв'язування громіздких або важко розв'язуваних проблем, які не можуть бути розв'язані алгоритмічним способом. Метою цього моделювання є знаходження задовольняючого рішення, яке найбільш наближається до оптимального. Докладніше евристичні моделі розглядаються в розділі 9.2.

Четвертим видом моделювання є *імітація* або *симуляція*. На противагу алгоритмізації та евристиці імітація забезпечує наочні результати шляхом проведення машинних експериментів. Метою імітації є відображення реальності в кількісному чи символічному вигляді. Вона включає повторення експерименту і описання характерних змін. Наприклад, імітація діяльності підприємства може містити такі змінні: середня кількість часу затримки конвеєра й кількість часу, який витрачається на його ремонт. Використовуючи імітаційну модель, можна змінювати попит на продукцію, сировину, кількість і вид продукції та вивчати вплив цих змінних на кількість часу, коли конвеєр зупиняється і знов починає працювати. За допомогою сучасних програм імітаційного моделювання ОПР можуть змінювати величини змінних і наочно бачити їх вплив за сценарієм «що ..., якщо...?».

Останнім типом методології є *аналітичне моделювання*. Воно полягає в поділі цілого на дрібніші частини, котрі досліджуються асоціативно з погляду їхньої поведінки, функцій і взаємозалежностей. Коли вже вивчена суть явища, то за допомогою аналітичних методів розв'язуються задачі з подібними змінними, які мають специфічні властивості у межах обмежень. Отже, ми можемо

використовувати специфічні рівняння залежно від поставленої мети. Коли явище не є повністю визначеним, що характерне, зокрема, для реальних бізнес-явищ, то аналітичні методи дають змогу розділити наявні проблеми на вибрані частини і визначити, які компоненти найбільше впливають на взаємозв'язки з іншими компонентами. Статистичний аналіз, особливо регресійний, є прикладом аналітичного моделювання.

Системи керування базою моделей у СППР

Керування моделями в СППР

Система керування моделями є одним із компонентів архітектури універсальної СППР. Функціями цієї системи є класифікація, організація і доступ до моделей, тобто ці функції аналогічні функціям системи керування базами даних. На рис. 5.7 наведено схему керування моделями, яка містить важливі елементи: базу моделей, систему керування базою моделей і керування діалогом як основну частину користувацького інтерфейсу.

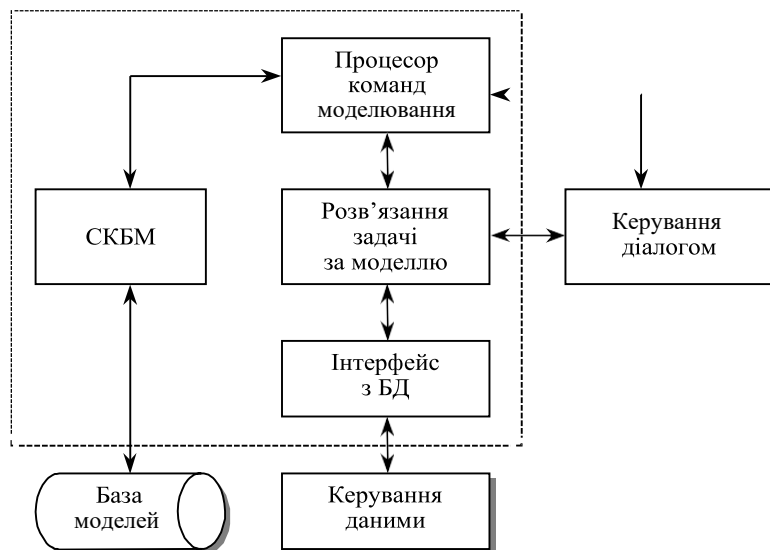


Рис. 1.7. Схема керування моделями в СППР

СППР безпосередньо забезпечує ОПР багатьма моделями. Через систему керування базою моделей (СКБМ) СППР забезпечує легкий доступ до моделей та допомагає їх використовувати. Очевидно, що база моделей є важливим аспектом цієї системи. Вона містить різноманітні статистичні, фінансові та управлінські моделі, які є важливими для розв'язування специфічних проблем, що

зустрічаються.

Основними функціями СКБМ є: створення нових моделей; каталогізація й оцінювання широкого діапазону моделей; поєднання компонентів моделей у базі моделей; виконання ряду загальних функцій управління.

Система керування базою моделей в СППР надає спектр можливостей користувачу, зокрема, забезпечує легкий доступ до моделей, допомагає усвідомлювати результати моделювання, забезпечує інтегрування моделей, дає змогу досліджувати чутливість рішень, надає інструментальні засоби керування моделями, уможливорює застосування зовнішніх моделей.

Програмне забезпечення СКБМ

Програмне забезпечення для СКБМ розроблене значно менше, ніж для СКБД чи користувацького інтерфейсу; наявним СКБМ притаманне розмаїття, а комерційні пакети СППР нерідко містять основні комбінації аналітичних методів розв'язування, статистичних пакетів та інших засобів моделювання. Повний комплект усіх сімей і підсімей методів моделювання зустрічається рідко, а частіше вони вмонтовані в систему процедури і засоби користувацького інтерфейсу. СКБМ для систематизованого формування, аналізу та інтерпретації моделей часто бувають спрощеними і обмеженими за своєю суттю. Перспективним напрямом створення ефективних СКБМ є структурне моделювання. Розроблено також кілька мов програмування досить високого рівня, спеціально пристосованих для створення елементів СКБМ для СППР.

Значна частина програмного забезпечення СКБМ має формат електронних таблиць. Зокрема, пакет Excel надає користувачеві можливість доступу до широкого діапазону математичних функцій та різних типів графіків; шляхом оброблення стовпців як послідовних часових інтервалів можна також моделювати систему змінних у часі. Цей пакет забезпечує розв'язання задач за допомогою аналізу рішень та інструментів динамічного програмування.

Зв'язок між користувацьким інтерфейсом і СКБМ

Важлива роль у системі керування моделями відводиться зв'язку користувацького інтерфейсу із СКБМ. Зокрема, проводяться роботи зі створення графічного інтерфейсу з вмонтованими засобами взаємодії на базі природної мови з метою надання користувачам змоги вводити команди і запитання зрозумілою для

них мовою. При цьому дуже важливо, аби система мала здатність автоматично вибрати найпридатніший модуль розв'язування задачі на підставі параметрів вхідного запиту і виду математичної моделі (наприклад, залежно від того, чи є поставлена задача оптимізаційною, чи ні). Це означає, що користувачеві потрібно вводити лише характеристики задачі або предметної галузі, а СКБМ автоматично та інтелектуально (тобто сама розпізнає задачу) здійснить вибір найпридатнішого методу чи інструменту. Також варто звернути увагу на роботи, в яких механізм інтерактивної взаємодії між користувачем і СКБМ розглядається з погляду об'єднання трьох факторів: мов структурування даних, запитів моделей і процесорів запитів природною мовою, зокрема, з орієнтацією на ключові слова процесора запитів для СКБМ, який інтегрує деякі характеристики цих компонентів. Прикладом мови запитів для структурованих моделей може бути система, в якій три операції — виконання, оптимізація й аналіз чутливості — утворюють критерій відносної повноти операцій.

Система на основі меню чи структурованої процедурної мови є нині найдоцільнішим варіантом у галузі інтерфейсів природною мовою (з урахуванням технологічних обмежень щодо природних мов у контексті комп'ютерного оброблення інформації). У варіанті оброблення структурованої природної мови запит вводиться в довільній формі, після чого він фільтрується з метою одержання низки ключових слів. Система має певні правила, згідно з якими проводиться опитування користувача, коли здобута послідовність ключових слів не збігається із записаним у пам'яті зразком. Як і в будь-якій іншій системі, у межах природної мови труднощі за створення такого інтерфейсу полягають у тому, аби забезпечити здатність системи правильно формулювати наміри користувачів і у відповідний спосіб реагувати на них.

Структурне моделювання

Загальна концепція структурного моделювання

У галузі архітектур і концепцій систем керування базами моделей останнім часом був досягнутий певний прогрес завдяки розвитку нового напрямку— *структурного моделювання*, яке може служити формальною математичною основою і обчислювальним середовищем для складання, подання і маніпулювання множиною моделей. Концепцію структурного моделювання запропонував 1987 року Артур Джеоффрион (Arthur M. Geoffrion), як відповідну реакцію на серйозні недоліки систем моделювання, доступних у 80-х роках XX ст. (<mailto:ageoffri@anderson.ucla>

edu). Цей підхід ґрунтується на ідеї, що будь-яка модель може бути розглянута, як сукупність чітких елементів, кожний з яких має визначення, яке, або є примітивним, або оснований на визначенні інших елементів моделі.

Структурне моделювання може бути розглянуте як використання ієрархічно організованого, розділеного і атрибутивного (attributed), направленного (без циклів) графа для відображення прикладу моделі або класу прикладів моделей. Кожна модель визначена з погляду впорядкованого ациклічного графа елементів (вузлів) і «викликів» (дуг). Головний вузол кожної дуги є «викликаючим елементом», а хвостовий — елементом, що «викликається».

Структурне моделювання може бути також розглянуте як система визначень для моделювання, що ґрунтується на таких п'яти засадних принципах:

1. **Кореляція.** Кожна взаємозалежність між елементами моделі має бути точно визначеною, щоб уникнути визначення за допомогою інших елементів.

2. **Ациклічність.** Граф моделі має бути ациклічним, щоб жоден елемент не міг бути безпосередньо або опосередковано визначений самим собою. Якщо в групі елементів існує циклічна залежність у визначеннях, то елемент не може бути визначений, оскільки буде існувати нескінченне поле для визначень.

3. **Класифікація.** Елементи моделі або вузли графа поділяють на п'ять типів, які називаються: *первинні об'єкти*, *складові об'єкти*, *атрибути/змінні*, *функції* і *тести*. Це забезпечує формальну семантичну основу для моделі і для полегшення програмної реалізації системи.

4. **Групування.** Елементи, які мають подібні визначення, об'єднуються в групи. Групування робить моделі прозорими і простими для керування. Це також скорочує розмірність моделі, оскільки кількість груп, які об'єднують у собі подібні елементи, не залежить від кількості самих елементів.

5. **Єрархія.** Групи єрархічно організовані.

У центрі концепції структурного моделювання знаходиться система визначень усіх елементів, які складають сутність поняття

«модель». У відповідності з Джеоффрином схема структурного моделювання базується на єрархічній побудові й має три рівні — *елементну структуру*, *родову структуру* і *модульну структуру* (*elemental structure, generic structure, modular structure*).

В **елементній структурі** модель розглядається як ряд дискретних елементів (елементів примітивних об'єктів), що має бути зображений у вигляді направленного графа елементів (вузлів) і «викликів» (дуг). Метою елементної структури є охоплення всіх детальних елементів у специфічному прикладі моделі. Елементна

структура моделі складається з окремих елементів з точною взаємозалежністю їх визначень.

Елементна структура будується на основі п'яти типів елементів:

1. *Примітивний об'єкт (PE)* використовується для відображення сутностей у проблемній сфері (або моделі) і він не має асоційованих значень.

2. *Складовий об'єкт (SE)* використовується для відображення визначення, яке посилається на один або більше вже певних об'єктів. Ці об'єкти можуть бути або примітивними, або складовими складних елементів (наприклад, зв'язок у транспортній системі між виробником і споживачем). Він також не має асоційованих значень.

3. *Атрибут (ATTR)* або *змінний атрибут (VATTR)* визначається в об'єднанні з деяким об'єктом або низкою об'єктів. Підтримка значень використовується для відображення окремих значень об'єкта, який визначений. Відмінність між атрибутом і змінним атрибутом полягає в тому, що в той час як атрибут використовується для відображення «коефіцієнтів даних» звичайних моделей, змінний атрибут використовується для відображення «змінних рішень».

4. *Функція (FUNC)* визначається як правило, що асоціюється з більш ніж одним об'єктом, який підтримує роботу зі значеннями, відповідно до правила.

5. *Тест (TEST)*. Цей перевіірочний елемент такий же як і функція, з тією лише відмінністю, що він може набувати одного з двох значень: «істинність» або «хибність». Використовується для визначення логічних аспектів моделювання.

Родова структура складається з груп елементів (родів, класів). Кожний рід у цій структурі об'єднує елементи одного типу (наприклад, ряд усіх примітивних елементів, які відображають центральну концепцію в предметній галузі), тобто родова структура використовується для групування подібних елементів. Вона визначається на елементній структурі як ряд розділів, по одному для кожного з п'яти типів елементів. Кожний із цих розділів називається *родом (класом)*, елементи яких однакові за винятком дрібниць.

Модульна структура використовується для згрупування класів у концептуальні одиниці, які називаються модулями відповідно до загальних або семантичних відношень, далі для об'єднання цих модулів у модулі вищого рівня і т. д. Модульна структура утворює єрархічну організацію другого рівня.

Головні переваги використання структурного моделювання такі:

— *Всезагальність*. Майже кожна схема моделі може бути визначена як структурна модель.

— *Зв'язок*. Структурна модель є дуже легкою для розуміння навіть для

користувачів з невеликими знаннями про проблеми і семантику дослідження операцій. Більшість мов, визначених на основі структурного моделювання, мають цю властивість.

— *Керованість*. Структурна модель є легкою для керування не тільки користувачами на рівні підтримки, але також і як частина системи. Структурні моделі легко об'єднуються для створення більших моделей.

Нині вже розроблено кілька пакетів програмного забезпечення структурного моделювання, зокрема, програмний продукт VMS, в якому поєднані концепції структурного та візуального моделювання.

Мова структурного моделювання

Метою структурного моделювання є забезпечення фундаменту для комп'ютерного середовища моделювання, яке використовує мову структурного моделювання SML (Structured Modeling Language), що підтримує цю моделюючу систему. SML вперше була розроблена Джоєфріоном, а після нього ряд інших авторів запропонували свої підходи до цієї проблеми. SML може бути подана чотирма рівнями. *Перший з них* містить прості визначені системи і моделі орієнтованих графів. *До другого рівня* належать більші комплексні їх розширення, моделі електронних таблиць, числові формули й моделі числення висловлень. *Третій рівень* охоплює математичне програмування і моделі числення предикатів з простою індексацією над множиною і декартовими добутками. Врешті, *четвертий рівень* містить розріджені версії вищезазначеного та реляційні і семантичні моделі бази даних.

Розглянемо використання SML на прикладі відомої *задачі про раціон (дієту)* або задачі про оптимальний склад суміші, відомої в англійській літературі як *Classical Feedmix Model*. Нагадаємо суть цієї задачі.

Задача про дієту або задача про раціон — це задача лінійного програмування, що полягає у визначенні такого раціону, який задовольняв би потреби людини або тварини в поживних речовинах за мінімальної загальної вартості продуктів, що використовуються. Це окремий і найпоширеніший приклад загальнішої задачі про оптимальний склад суміші.

Задача визначення оптимального раціону для людини складна, оскільки доводиться враховувати багато додаткових чинників, що не завжди можна формалізувати: смакову прихильність, різноманітність блюд тощо. Однак у тваринництві визначення раціонів для худоби за допомогою задачі лінійного програмування сьогодні не просто реальне, але й необхідне. Досвід свідчить, що

годілля худоби за раціонами, розрахованими цим методом, дає істотну економію. Наприклад, у США такі раціони використовують багато фермерів. Це не означає, зрозуміло, що кожний з них сам розв'язує задачі лінійного програмування: для різних регіонів країни видаються довідники раціонів годівлі тварин, ураховуючи місцеві особливості й можливості, породи худоби тощо.

Наведена нижче схема SML (третій рівень) описує загальну структуру класичної моделі раціону (feedmix), а приклади SML для елементних детальних таблиць, які конкретизують елементи моделі, разом зі схемою визначають специфічну (конкретну) модель feedmix.

&NUT_DATA NUTRIENT DATA (дані про поживні речовини)

NUTRi /pe/. Це список поживних речовин (**NUTRIENTS**). **MIN (NUTRi) /a/ : Real+** Для кожної поживної речовини

(**NUTRIENT**) є *MINIMUM DAILY REQUIREMENT* (мінімальна щоденна потреба) — одиниць на день на тварину.

&MATERIALS MATERIALS DATA (дані про матеріали)

MATERIALm /pe/ Це список кормів (**MATERIALS**), які можна використовувати для годівлі.

UCOST (MATERIALm) /a/ Кожний вид корму (**MATERIAL**) має *UNIT COST* (питому ціну) — дол. за фунт корму.

ANALYSIS (NUTRi, MATERIALm) /a/ : Real+ Для кожної комбінації **NUTRIENT-MATERIAL** (поживна речовина— продукт) проводиться аналіз (**ANALYSIS**) вмісту поживної речовини у фунті продукту.

Q (MATERIALm) /va/ : Real+ **QUANTITY** Має визначатися щоденна кількість (у фунтах) кожного виду корму (**MATERIAL**) на тварину.

NLEVEL (ANALYSISi, Q) /f/ ; @SUMm (ANALYSISim *

Qm) Як тільки кількості (**QUANTITIES**) кормів вибрані, визначається рівень поживності (**NUTRITION LEVEL**) — одиниць щоденно на тварину для кожної поживної речовини, обчислюваний за допомогою аналізу (**ANALYSIS**).

T:NLEVEL (NLEVELi, MINi) /t/ ; NLEVELi >= MINi. Для кожної поживної речовини проводиться зіставлення (тестування) (**NUTRITION TEST**), щоб визначити, чи рівень поживності кормів (**NUTRITION LEVEL**) не менший, ніж мінімальна щоденна потреба (**MINIMUM DAILY REQUIREMENT**).

TOTCOST (UCOST, Q) /f/ ; @SUMm (UCOSTm * Qm). Це загальна сумарна вартість раціону (**TOTAL COST**) у доларах на день на тварину, яка обчислюється за вибраними їх кількостями (**QUANTITIES**).

Структура й послідовність елементних детальних таблиць залежить процедурно від схеми. Кожна таблиця має назву, тобто ім'я стовпця, яке зазвичай збігається з іменем роду, і має рядок для кожного елемента відповідного виду. Таблиці даних з конкретизацією схеми SML для задачі про раціон мають такий вигляд:

NUTR		
<i>NUTR</i>	<i>Інтерпретація</i>	<i>MIN</i>
P	Protein <i>протеїн</i>	16
C	Calcium <i>кальцій</i>	4

MATERIAL		
<i>MATERIAL</i>	<i>Інтерпретація</i>	<i>UCOST</i>
std	Standard Feed <i>Стандартний корм</i>	1.20
add	Additive <i>добавка</i>	3.00

ANALYSIS		
<i>NUTR</i>	<i>MATERIAL</i>	<i>ANALYSIS</i>
P	std	4.00
P	add	14.00
C	std	2.00
C	add	1.00

Q	
<i>MATERIAL</i>	<i>Q</i>
std	2.00
add	0.50

NLEVEL		
<i>NUTR</i>	<i>NLEVEL</i>	<i>T:NLEVEL</i>
P	15.00	FALSE (хибність)
C	4.50	TRUE (істина)

TOTCOST
TOTCOST 3.90

Дамо стислі пояснення синтаксису мови структурного моделювання SML стосовно наведеного прикладу. Схеми моделі організовуються як дерево параграфів (абзаців), виходами яких є роди, а внутрішні вузли є модулями. Виділена напівжирним шрифтом частина кожного абзацу є формальним визначенням роду чи виду (genus) або модуля залежно від контексту, а решта складається з документальних коментарів про формальну частину, які є неформальними за винятком домовленостей про використання підкреслення і верхнього регістра. Формальне визначення параграфа (абзацу) роду починається з імені роду (наприклад — NUTRi); вставного твердження визначеної залежності (якщо така є, наприклад UCOST, Q); розмежованого слешем (/) твердження типу роду (наприклад, /va/ — змінний атрибут, /f/ — функція, /t/ — тест); оголошеного двокрапкою твердження типу даних (наприклад, Real+ — дійсні додатні числа), якщо це атрибутний рід (genus); і оголошений крапкою з комою математичний вираз, який називають генеруючим правилом, якщо це функція (наприклад, @SUMm (UCOSTm * Qm) — сума добутків) або тестовий рід (genus) (наприклад, NLEVELi >= MINi — перевірка нерівності).

Формальне визначення модульного параграфа складається тільки з імені. Зверніть увагу на те, що схема завжди описується незалежно від будь-якої проблеми або завдання, яке могло б формулюватися в такому разі. Загальне завдання, яке описане вищезазначеною схемою, — знайти такі значення для всіх елементів Q, що всі елементи T: NLEVEL оцінюються як істина і значення елемента TOTCOST мінімальне.

На рис. 1.8 наведений родовий граф, який відображає структуру моделі про раціон, і який пов'язаний з вищезазначеною схемою. Він зображає визначену залежність рівнів родів.

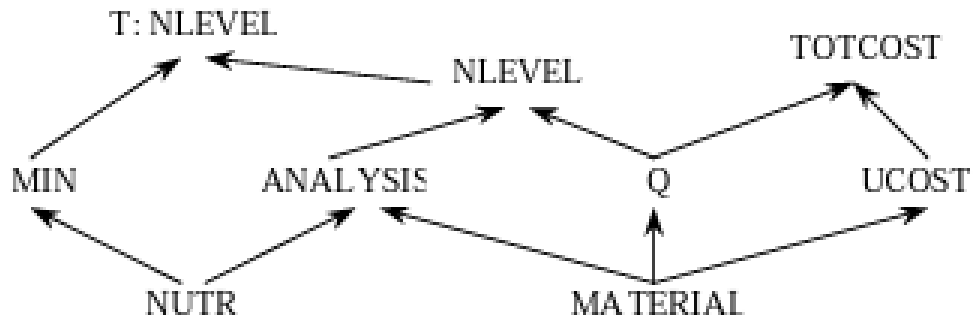


Рис. 1.8. Родовий граф багатоеlementної моделі задачі про раціон:
 Q — склад добового раціону; NUTR — поживні речовини; MIN —
 мінімальна щоденна потреба в поживних речовинах;
 TOTCOST — загальна вартість раціону; UCOST — ціна за одиницю
 (питома ціна) корму; NLEVEL — рівень поживності; T:NLEVEL —тестування
 поживності; MA TERIAL —корми, що входять до раціону; ANALYSIS —
 проведення аналізу кормів на поживність

SML-базові системи моделювання можуть мати специфічні властивості, яких часто бракує в системах моделювання загальноприйнятішого проекту.

Використання структурного моделювання для створення СКБМ

Структурна модель може відображати як одночасність, так і рекурсію, і здатна містити дійсно динамічні моделі з дискретизованим часом. Можна також використовувати й безперервний час, якщо вважати значення атрибутів і функціональних елементів функціями часу. До задач, що розв’язуються за допомогою структурних моделей, можна віднести оптимізацію, оброблення незапланованих запитів користувачів, пошук значень вибраних атрибутів, одержання логічних висновків за рахунок використання системи як механізму виведення. Такі широкі різнобічні дії з розв’язання завдань оброблення інформації охоплюють досить великий діапазон потреб і функцій підтримки прийняття рішень у СКБМ і одночасно забезпечують незалежність подання моделі і розв’язків, одержаних на її основі.

Використання структурного моделювання для цілей проектування і розроблення СКБМ включає:

- низхідне проектування моделі, що створює можливість поетапного вдосконалення за рахунок єрархічної картини зони моделювання;

— інтегроване моделювання як за функціями (наприклад, виробництво і розподіл), так і в географічному та часовому аспектах (наприклад, планування і побудова календарних графіків);

— досягнення цілей комунікації і документування, оскільки спрощена версія схеми моделі досить корисна для отримання повідомлень звичайною мовою і, як наслідок, для комунікації з користувачами.

Родовий граф (подібний зображеному на рис. 1.8) може виявитись досить корисним щодо комунікації: є можливість швидко безпосередньо побачити фактичну структуру моделі.

Управління поштою (повідомленнями) в СППР

Традиційно фахівці вважали поштову систему, навіть електронну пошту, додатковою функцією СППР. Тобто розробники систем підтримки прийняття рішень розуміли, що особа, яка приймає рішення, ймовірно, має системи доставлення електронної пошти, але такі системи ігноруються за проектування і розроблення СППР. Для використання електронної пошти ОПР спочатку мали б припинити роботу з СППР. Навіть коли з'явилася технологія «роботи з вікнами» прикладних програм, яка дала змогу користувачам легко переміщуватися в прикладну програму електронної пошти, СППР і електронна пошта існували все ще як незалежні програми. ОПР не могла легко пересилати документи, графіку чи текст безпосередньо з СППР або отримувати такі, щоб використовувати всередині СППР.

У нинішньому електронному середовищі безшовна інтеграція СППР з системою електронної пошти означає серйозні обмеження на інформацію та аналізи, доступні ОПР. Усе більша кількість менеджерів вважає електронну пошту продуктивно розширювальним інструментом. Інтерфейс із системою електронної пошти уможливорює ОПР більший доступ до груп обговорення, баз даних Інтернету, інших електронних даних та інструментальних засобів для прийняття рішень. Ці ресурси можуть розширювати діапазон доступної інформації, а також за деяких обставин забезпечувати доступ до новіших даних. Використання електронної пошти може допомагати ОПР зв'язуватися з колегами, щоб обговорити інформацію та дослідження, а також встановити загальноприйнятну перспективу рішень. Це може, у свою чергу, покращити зв'язок ОПР з підлеглими і сформувати підтримку для вибору, що є важливими кроками в процесі прийняття

рішень.

Розглянемо можливості, які може забезпечити ОПр доступ до електронної пошти. По-перше, електронна пошта може дати змогу ОПр легко надсилати і отримувати інформацію від колег. Така гнучкість особливо важлива, коли ОПр мають спілкуватися з колегами в інших країнах. Інакше ОПр мають обмаль часу щодня для того, щоб увійти в контакт з іноземними колегами, і не можуть підтримуватистосунки на оптимальному рівні.

Спілкування між ОПр, зазвичай складніше, ніж простізапитання і відповіді, що можуть бути реалізовані з застосуванням простого шаблону пошти. Часто вони мають формувати документи і проводити дослідження на основі аналізу чи пов'язаних записів. Якщо електронна пошта інтегрована в СППР, то вони можуть легко вкладати документи, електронні таблиці, і навіть графіки, що збагачує зв'язок. Багато із сьогоднішніх комерційних програм електронної пошти мають можливість вкладати документи і записи в електронній пошті. Якщо користувач часто вкладає окремі документи чи типи документів, то він може у відповідний спосіб налаштовувати панель інструментів для зручнішої реалізації своїхзавдань.

Використання електронної пошти для щоденного зв'язку, який традиційно проводиться у формі зустрічей, телефонних дзвінків чи поштового обслуговування, створює в людей звичку використовувати цей засіб комунікацій. З причин, які не очевидні, більшість людей мають тенденцію відповідати на електронну пошту легше і швидше, ніж використовувати якісь інші типи зв'язку. Так само люди із задоволенням використовують електронну пошту для передавання повідомлень, які інакше не можуть бути передані, особливо особам вищого рангу. Електронна пошта забезпечує зв'язок між особами, які, зазвичай, не взаємодіють між собою. Крім цього, багато людей надають своїм повідомленням електронною поштою більш персональний характер ніж записам.

Крім прямого використання електронної пошти користувачіСППР можуть використовувати електронні дискусійні групи, щоб отримати інформацію чи оцінки від розширеної групи колег.Відкриті дискусійні групи доступні в Інтернеті майже на будь-

яку тему, яку тільки можна уявити. Деякі дискусійні групи у США містять промислові теми, такі як музична промисловість, управління готелями та ресторанами, або навіть автоматизовані системи відкачування рідини. Є також дискусійні групи, які цікаві для ОПр тим, що можуть підтримувати специфічні види завдань — адміністраторів мереж, інженерів чи групи обговорення технічних стандартів. Нарешті, є деякі загальні в галузі прийняття рішень дискусійнігрупи,

такі як загальне управління якістю (TQM), керованої охорони здоров'я і виконавчі групи.

ОПР можуть також ініціювати дискусійні групи, складені зі службовців їх корпорацій, відділів чи проектних груп, або колег з подібними інтересами та обов'язками з інших корпорацій. Так само дискусійні групи могли б використовуватися для пропозицій щодо процедурних чи стратегічних змін і (або) для ідентифікації проблеми важливості певних клієнтів чи підлеглих. Користувачів цих закритих дискусійних групах можуть бути більш наближеними до інформації та отримувати підтримку, тому що вони саме ті, хто буде читати повідомлення; вони також не мають хвилюватися стосовно відкритості приватної інформації чи стратегічних планів корпорації. Крім того, закриті дискусійні групи мають спільну мету і відповіді на запитання можуть допомогти наблизитися до цієї спільної мети.

Деякі пакети електронної пошти, доступні на сьогодні, значно полегшують такі обговорення. Наприклад, поштова система DaVinci містить інформаційне табло для групових повідомлень. За використання такого засобу ОПР можуть формувати групові обговорення, легко і швидко відповідати на запитання чи запити, як тільки-но вони з'являються. Важливішим є те, що пакети типу DaVinci мають здатність відстежувати нитки повідомлень електронної пошти. Якщо ОПР хоче слідкувати за діалогом зі специфічної тематики, то вона може знайти початкове повідомлення, відповіді на повідомлення, і навіть відповіді на відповіді повідомлень легко, навіть якщо було багато інших поштових повідомлень у той же період. Тому ОПР не відволікається на інші теми та може зосередитися на проблемі, що розв'язується.

Ще один спосіб використання електронної пошти здійснюється шляхом перегляду певного середовища, щоб ідентифікувати проблеми й можливості. Наприклад, дискусійні групи чи групи новин можуть допомогти ОПР виявити нові ідеї. Запитання, поставлені іншими користувачами відносно нових методів або процедур, можуть дійти до уваги ОПР, які можуть розглянути їх як стратегічні можливості чи використати ідеї для розв'язування наявних проблем. У такий же спосіб ОПР з міжнаціональних корпорацій, чи ті, хто беруть участь у багатонаціональних переговорах, можуть контролювати поточні події в окремих країнах, шукати зміни, які можуть потребувати змін у політиці чи планах. Так само і внутрішні дискусійні групи могли б використовуватися для виявлення проблем чи ідей.

Дискусійні групи — не єдиний шлях, яким ОПР можуть використовувати електронні комунікації. Існує ряд інформаційних служб, на повідомлення яких

можна підписатися в Інтернеті, що забезпечують, наприклад, своєчасні і технологічно пов'язані промислові новини. Від них ОПР можуть отримати ряд новин, які пов'язані з вибором, що розглядається. Цей сервіс також надає фінансову інформацію, курси акцій та інші матеріали, які можуть бути цікавими для ОПР.

Не обов'язково підписуватися на всі послуги. Недавні зміни в протоколі Інтернету значно збільшили можливості отримання й пошуку електронних баз даних. ОПР можуть шукати всі електронні бази даних за окремими темами з незначними зусиллями. Вони можуть переглядати загальні телефонні книги, офіційні повідомлення для друку та щоденні звіти з усього світу.

Розглянемо, наприклад, систему infoSage від IBM. Ця система забезпечує інтелектуальне фільтрування електронної пошти, яка поступає в систему. Після переривання та читання повідомлення, infoSage пробує класифікувати його, використовуючи певні правила. Ці правила можуть відбивати важливість відправника інформації, самої інформації чи джерела повідомлення. Програма може виконувати такі дії: відбирати повідомлення з оцінюванням пріоритету, зберігати повідомлення в папці, відкидати повідомлення, переглядати їх тощо.

Подібні системи у складі СППР можуть використовуватися для фільтрації новин чи пошти, інформації дискусійних груп. Зокрема, користувач має вибрати зі списку те, що краще описує його потреби. Він може також доповнювати список специфічними ключовими словами, які можуть зустрічатися в новинах. Нарешті, користувач може розмістити за важливістю теми списку, щоб відбити свої відносні інтереси. Потім, з регулярними інтервалами, система скануватиме сервіси новин і шукатиме інформацію, яка відповідає інтересам користувача. Після цього статті будуть організовані в групи з предметними заголовками та розміщені за важливістю відповідно до інтересів користувача.

Пошук здійснюється за допомогою технології клієнт/сервер та браузерів Інтернету, таких як Netscape Navigator. Використовуючи такі інструментальні можливості, ОПР можуть отримати доступ до даних, розподілених і підтримуваних на ЕОМ у всьому світі. Альтернативно вони можуть зосередити пошук на спеціальних індексах (пошукових сервісах), щоб знайти всі директорії, в яких використовуються певні ключові поля. Так само ОПР можуть використовувати засоби пошуку, щоб здійснити повний текстовий пошук за всіма документами і вивести список тих, які містять певні ключові слова.

Багато бібліотек починають забезпечувати електронні повнотекстові версії матеріалів-посилань і друкують матеріали, які можуть бути доступними через WWW. Нарешті, деякі професійні групи почали забезпечувати інтерактивний

доступ до журналів чи резюме статей. Урешті-решт можливості нескінченні для тих, хто знає, як доступитися до системи.

Методи для адресування проекту поштової управлінської системи подібні до тих, які використовуються для управління даними й моделями. Зокрема, якщо ОПР мають доступ до адекватної системи електронної пошти, то розробники СППР не повинні перепроєктовувати систему, а скоріше, вони мають забезпечити зв'язки між цією системою електронної пошти і СППР. Якщо, однак, ОПР не мають доступу до адекватної системи електронної пошти, було б доцільно включити цю можливість у повний проект.

Адекватна система електронної пошти — це та, яка має можливості, що роблять її корисною для прийняття рішень. Такі системи мусять мати доступ до стандартних можливостей електронної пошти, типу здатності індексувати повідомлення та відображати ці індекси, а також мати здатність реєструвати повідомлення, видаляти їх, направляти іншим зацікавленим особам, та відповідати на них. Крім того, вони мають містити легко і автоматично доступну адресну книгу осіб.

Щоб бути корисною в середовищі СППР, система електронної пошти потребує додаткових можливостей, які полегшують чи хоча б не перешкоджають використувувати системи електронної пошти для підтримки рішень. Одна з таких можливостей — автоматичне повідомлення системи користувачеві про доступність електронної пошти, незалежно від того, чи використовується якийсь інший додаток. Якщо автоматичне повідомлення не забезпечується, то користувач вимушений зупинити те, що він робить для того, щоб перевірити електронну пошту. За цих умов ОПР, імовірно, не будуть використовувати електронну пошту, або, у крайньому разі, не будуть використовувати її ефективно. Звичайно, ОПР мають бути здатними відключати функцію автоматичного повідомлення, коли вони не хочуть відриватися від виконання поточних завдань.

З цією можливістю пов'язана необхідність забезпечувати систему фільтрації повідомлень, оскільки автоматичне повідомлення без механізму фільтрування може зробити систему електронної пошти більше дратуючою, ніж корисною. Зрозуміло, що швидке повідомлення про нетермінову інформацію не краще, ніж взагалі не мати його.

Проте якщо система електронної пошти повідомляє користувача тільки тоді, коли повідомлення отримано від окремих осіб чи дискусійних груп, то це може бути корисною інформацією. Так само, якщо фільтруюча система може читати повідомлення і переривати роботу лише в тому разі, якщо надійшла електронна

пошта відносно певної теми чи переліку тем, то це буде корисно для ОПР. Крім того, така система фільтрування могла б відрізняти повідомлення, що є відповідями на запити ОПР, від тих, які є новими, щоб надати пріоритети повідомленням про відповіді на поставлені користувачем запитання.

Якщо система електронної пошти здатна визначити джерело повідомлення, порівнявши його з переліком джерел визначеного користувача, і (або) визначити тему повідомлення та зіставити її з низкою тем визначеного користувача, то вона буде «знати», які повідомлення важливі для ОПР, і відповідно до цього переривати її роботу.