

**ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ З
ДИСЦИПЛІНИ «ЛЮДИНО-МАШИННА
ВЗАЄМОДІЯ»**

Лекція №1. Основні поняття про людино-машинну взаємодію

План:

1. Мета і задачі дисципліни.
2. Різновиди людино-машинних систем. Роль та місце людини в контурі людино-машинних систем.
3. Підхід програмної інженерії до людино-машинної взаємодії.
4. Поняття про юзабіліті.

Конспект лекції

1.1 Мета і задачі дисципліни

МЕТА ДИСЦИПЛІНИ: вивчення понять, принципів і засобів, що використовуються при створенні людино-машинних інтерфейсів програмних систем.

ЗАДАЧІ ВИВЧЕННЯ ДИСЦИПЛІНИ і її зміст визначені відповідно до місця та значення дисципліни за структурно-логічною схемою, передбаченою освітньо-професійною програмою підготовки бакалавра з напрямку «Програмна інженерія», і охоплює всі змістовні модулі, визначені анотацією для мінімальної кількості годин, передбачених стандартом.

У результаті вивчення дисципліни “Людино-машинна взаємодія” студенти повинні **ЗНАТИ:**

- особливості сприйняття інформації людиною;
- пристрої і режими діалогу;
- питання комп'ютерного подання і візуалізації інформації;
- парадигми і принципи взаємодії людини з комп'ютерним середовищем;
- критерії оцінки корисності діалогових систем;
- тенденції розвитку користувацьких інтерфейсів.

УМІТИ:

- будувати і описувати взаємодію з комп'ютерним середовищем в заданій проблемній області;
- користуватися бібліотеками елементів управління діалогом, програмами підтримки розробки користувацьких інтерфейсів;
- створити середовище, описати події і реалізувати інтерактивну систему.

Об'єктом вивчення дисципліни є людино-машинна взаємодія в широкому значенні.

Предметом дисципліни є людино-комп'ютерна взаємодія.

Програма курсу розроблена на основі типової програми, рекомендованої науково-методичною комісією з програмної інженерії. Вона розрахована на 108 год. (три кредити).

Дисципліна складається з двох модулів. Протягом викладання дисципліни передбачається проведення 2-х колоквиумів.

Зауваження 1.1. Об'єкт дослідження — це процес або явище, що породжує проблемну ситуацію і обране для вивчення.

Предмет дослідження міститься в межах об'єкта. Об'єкт і предмет дослідження, як категорії наукового процесу, співвідносяться між собою як загальне і часткове. В об'єкті виділяється та його частина, яка є предметом дослідження. Саме на нього спрямована основна увага дисертанта, оскільки предмет дослідження визначає тему дисертаційної

праці, яка визначається на титульній сторінці як її назва.

1.2 Різновиди людино-машинних систем. Роль та місце людини в контурі людино-машинних систем

Людино-машинний інтерфейс (ЧМІ) (англ. Human machine interface, HMI) - широке поняття, що охоплює інженерні рішення, що забезпечують взаємодію оператора з керованими їм машинами.

Створення систем людино-машинного інтерфейсу тісно пов'язане з поняттями ергономіка і юзабіліті.

Зауваження 1.2. Інтерфейс (від англ. Interface - поверхня розділу, перегородка) - сукупність засобів, методів і правил взаємодії (управління, контролю і т. Д.) між елементами системи.

Цей термін використовується в багатьох областях науки і техніки. Його значення відноситься до будь-якого сполученню взаємодіючих сутностей (як природничо-наукових, так апаратних і людино-машинних). Під інтерфейсом розуміють не тільки пристрої, але і правила (протокол) взаємодії цих пристроїв.

Проектування ЧМІ включає в себе:

- створення робочого місця: крісла, стіл, або пульта управління, розміщення приладів і органів управління (відповідністю всього цього фізіології людини займається Ергономіка), освітлення робочого місця і, можливо, мікроклімат.

- далі розглядаються взаємодія оператора з усіма органами управління: їх доступність і необхідних зусиль, ефективності та швидкості доступу, узгодженість (несуперечність) керуючих впливів (в т. Ч. Т. Н. «Захист від дурня»), розташування дисплеїв і розміри написів на них (все це входить в сферу Юзабіліті)

Однією з найбільш складних завдань є створення ЧМІ робочих місць складних машин з безліччю органів управління: пілотів літака і космічних кораблів.

У промислових умовах ЧМІ найчастіше реалізується з використанням типових засобів: операторських панелей, комп'ютерів і типового програмного забезпечення.

- ГОСТ Р МЕК 60447-2000 Інтерфейс людино-машинний. Принципи приведення в дію.

- ГОСТ Р МЕК 60073-2000 Інтерфейс человекомашина. Маркування та позначення органів управління і контрольних пристроїв. Правила кодування інформації.

Ергономіка (від грец. "Εργον - робота і νόμος - «закон») - в традиційному розумінні - наука про пристосування посадових обов'язків, робочих місць, обладнання та комп'ютерних програм для найбільш безпечного та ефективного праці працівника, виходячи з фізичних і психічних особливостей людського організму.

Більш широке визначення ергономіки, прийняте в 2010 році Міжнародною Асоціацією ергономіки (IEA), звучить так: «Наукова дисципліна, що вивчає взаємодію людини та інших елементів системи, а також сфера діяльності по застосуванню теорії, принципів, даних і методів цієї науки для забезпечення благополуччя людини і оптимізації загальної продуктивності системи».

1.3 Підхід програмної інженерії до людино-машинної взаємодії

Людино-комп'ютерна взаємодія (HCI) - це вивчення, планування і розробка взаємодії між людьми (користувачами) і комп'ютерами. Найчастіше його розглядають як сукупність науки про комп'ютери, біхевіоризму, проектування та інших областей дослідження. Взаємодія між користувачами і комп'ютерами відбувається на рівні призначеного для користувача інтерфейсу (або просто інтерфейсу), який включає в себе програмне і апаратне забезпечення; наприклад, образи або об'єкти, які відображаються на екранах дисплеїв, дані, отримані від користувача за допомогою апаратних пристроїв

введення (таких як клавіатури і миші) та інші взаємодії користувача з великими автоматизованими системами, такими як повітряне судно і електростанція.

Зауваження. Інтерфейс користувача — сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача; у графічних системах інтерфейс користувача реалізовується багатовіконним режимом, змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їхнім розташуванням, сортуванням елементів вікон, гнучкими налаштуваннями як самих вікон, так і окремих їхніх елементів (файли, папки, ярлики, шрифти тощо), доступністю багатокористувацьких налаштувань. Дружній інтерфейс користувача — засіб зручної взаємодії користувача з інформаційною системою.

Асоціація обчислювальної техніки розглядає взаємодію людини і комп'ютера як «дисципліну, що займається проектуванням, оцінкою і здійсненням роботи інтерактивних обчислювальних систем для використання людиною, а також вивченням процесів, що відбуваються.» Важливим аспектом людино-комп'ютерного взаємодії є забезпечення задоволення користувачів (див. Computer user satisfaction) .

У зв'язку з тим, що людино-комп'ютерна взаємодія вивчається як з людського боку, так і з комп'ютерною, то знання, отримані в ході дослідження, спираються як на людський фактор, так і на комп'ютерний. З комп'ютерної боку важливі технології комп'ютерної графіки, операційних систем, мов програмування і середовища розробки. З людського боку, теорія комунікації, графічне і виробниче проектування, лінгвістика, соціологія, когнітивна психологія і такі людські фактори як задоволення користувачів. Також має значення інженерія та проектування. Завдяки міждисциплінарного характеру людино-комп'ютерного взаємодії, люди з різним рівнем підготовки вносять вклад в його успіх. Іноді людино-комп'ютерна взаємодія називають як людино-машинне взаємодія, так і комп'ютерно-людське взаємодія.

Важливим критерієм є увагу до людино-комп'ютерного взаємодії, тому що погано розроблені інтерфейси можуть стати причиною багатьох непередбачених проблем. Класичним прикладом цього є аварія на АЕС Три-Майл-Айленд, де в ході розслідування було виявлено, що, принаймні, часткову відповідальність за катастрофу несе на собі проектування інтерфейсу. Подібним чином, аварії в авіації виникали внаслідок рішення виробників використовувати нестандартні повітряні прилади та / або розташування штурвала. Хоча передбачалося, що нові конструкції більш досконалі щодо основного людино-комп'ютерного взаємодії, пілотам було притаманне «стандартне» розташування і, таким чином, концептуально хороша ідея, не спричинила бажані результати.

цїлі

Основним завданням людино-комп'ютерного взаємодії є поліпшення взаємодії між людиною і комп'ютером, роблячи комп'ютери більш зручними (юзабельними) і сприйнятливими до потреб користувачів. Зокрема, людино-комп'ютерна взаємодія займається:

- методологією і розвитком проектування інтерфейсів (тобто, виходячи з вимог і класу користувачів, проектування найкращого інтерфейсу в заданих рамках, оптимізація під необхідні властивості, такі як здатність до навчання і ефективність використання);

- методами реалізації інтерфейсів (наприклад, програмні інструментарії, бібліотеки і раціональні алгоритми);

- методами для оцінки та порівняння таких інтерфейсів;

- розробкою нових інтерфейсів і технологій взаємодії;

- розвитком описових і прогнозованих моделей, і теорією взаємодії;

Довгостроковим завданням людино-комп'ютерного взаємодії є розробка системи, яка знизить бар'єр між людської когнітивної моделлю того, чого вони хочуть досягти і розумінням комп'ютера поставлених перед ним завдань.

Фахівці людино-комп'ютерного взаємодії - це, як правило, розробники, що займаються практичним застосуванням методик розробки до реальних всесвітнім

проблемам. Їх робота, найчастіше, обертається навколо розробки графічних- і веб-інтерфейсів.

Дослідники людино-комп'ютерного взаємодії займаються розвитком нових методик проектування, проведенням експериментів з новими апаратними пристроями, створенням прототипів нових систем програмного забезпечення, вивченням нових парадигм для взаємодії і розвитком теорій і моделей взаємодії.

Створення якісного людино-комп'ютерного інтерфейсу, який можна назвати точкою зв'язку між людиною і комп'ютером, є кінцева мета вивчення людино-комп'ютерного взаємодії.

4 Поняття про юзабіліті

Юзабіліті (англ. Usability - дослівно «можливість використання», «здатність бути використаним», «корисність») - поняття в мікроергономіке, що позначає підсумковий рівень зручності предмета для використання в заявлених цілях.

Термін має зв'язок з поняттям «ергономічність», але на відміну від останнього менше асоціюється з технічною естетикою, з зовнішнім виглядом і більш прив'язаний до утилітарності «юзабельного» об'єкта. Український аналог – зручність використання.

Офіційне визначення

Міжнародний стандарт ISO 9241-11 визначає юзабіліті як «ступінь, з якою продукт може бути використаний певними користувачами при певному контексті використання для досягнення певних цілей з належною ефективністю, продуктивністю і задоволеністю» (англ. «The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use »). При цьому відносна важливість усіх трьох аспектів визначається цим самим контекстом.

Як і багато іноземних терміни, які в новітній час увійшли в ужиток на «пострадянському просторі», слово «юзабіліті» нерідко використовується без урахування його значення в початковому англійською і поза зв'язком з міжнародними стандартами інженерної термінології. Як правило це відбувається через те, що ігнорується ключовий для контексту момент «визначеності». Тому часто відбувається так, що «інтуїтивна зрозумілість» для розробників не є такою для кінцевих користувачів.

При розробці користувацьких інтерфейсів словом юзабіліті позначають загальну концепцію їх зручності при використанні програмного забезпечення, логічність і простоту в розташуванні елементів управління. Однак при цьому нерідко відбувається підміна понять - утилітарних естетичними. Тоді чітко не визначений в цілях і не обгрунтована в деталях корисність конкретного рішення для користувача, вважається сама собою зрозумілою, яка витікає з оригінальності зовнішнього вигляду.

Термін «юзабіліті» можна розглядати як синонім слова «ергономічність» з тією різницею, що остання визначає мінімальність конкретних фізичних зусиль при користуванні річчю, а перша - кінцеву сумарну ступінь зручності, міру інтелектуального зусилля необхідного для отримання корисних якостей цієї речі і швидкість досягнення позитивного результату при управлінні нею. Тому стосовно таких продуктів, як побутова електроніка або засоби зв'язку, частіше використовують поняття «ергономічність» щодо форми, компонування вузлів і ваги, наприклад, а «юзабіліті» - стосовно зрозумілості елементів управління і режимів роботи, до кількості і необхідності функцій.

У більш широкому значенні термін «юзабельність» вживається як зручність користування (наприклад, для механічних пристроїв та інструментів (таких як дверна ручка або молоток) ергономічність форми буде підвищувати юзабельність речі (тобто «зручність застосування», «дружність і простоту при користуванні », «практичність » і взагалі «застосовність »)).

Юзабіліті має серйозне значення для показників конверсії в електронній комерції. Однак юзабіліті означає не тільки покращене візуальне керівництво або поліпшену

ієрархію сайту. Це також означає і більший контакт з потенційним покупцем за допомогою професійно зробленого серйозного дизайну, піднесення вірною інформації тоді, коли вона потрібна.

Підсумок

ЗНАТИ: загальні поняття про людино-комп'ютерний інтерфейс.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людино-комп'ютер.

Контрольні питання і вправи

1. Що таке ергономічність?
2. Дайте визначення юзабіліті.
3. Охарактеризуйте відмінності поняття «ергономічність» та «юзабельність».
4. Наведіть визначення юзабіліті.
5. Що таке інтерфейс?

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Влад В. Головач. Дизайн пользовательского интерфейса². Искусство мыть слона — 2009. — 94 с.
2. Якоб Нильсен, Хоа Лоранжер. Web-дизайн: удобство использования Web-сайтов Prioritizing Web Usability — М.: «Вильямс», 2007. — 368 с. — ISBN 0-321-35031-6.
3. Стив Круг. Веб-дизайн: книга Стива Круга или «не заставляйте меня думать!» Don't make me think! — М.: Символ-плюс.
4. Стив Круг. Как сделать сайт удобным. Юзабилити по методу Стива Круга = Rocket Surgery Made Easy: The Do-It-Yourself Guide to Finding and Fixing Usability Problems — СПб.: Питер, 2010. — С. 208. — ISBN 978-5-49807-515.
5. Джеф Раскин. Интерфейс: новые направления в проектировании компьютерных систем — Символ-Плюс. — 2004. — ISBN 5-93286-030-8.
6. Алан Купер. Психбольница в руках пациентов или Почему высокие технологии сводят нас с ума и как восстановить душевное равновесие = The Inmates are Running the Asylum — Символ-Плюс, 2004. — ISBN 5-93286-071-5.
7. Магазанник В. Д., Львов В. М. Человеко-компьютерное взаимодействие: Учебное пособие для вузов — Тверь: Триада, 2005. — С. 200. — ISBN 5-94789-119-0.
8. Гарретт Д. Веб-дизайн: книга Джесса Гарретта. Элементы опыта взаимодействия. The Elements of User Experience: User-Centered Design for the Web — Символ-Плюс, 2008. — С. 192. — ISBN 5-93286-108-8.

Глосарій, персоналій

Людино-машинний інтерфейс (ЛМІ) (Human machine interface, HMI); людино-комп'ютерний інтерфейс (Human computer interface, HCI); юзабіліті (Usability); ергономіка (від грец. "Εργον - робота і νόμος - «закон»).

Лекція №2. Компоненти інтерфейсу користувач-комп'ютер. Стратегії розробки інтерфейсу

План:

1. Інтерфейс користувач-комп'ютер та його зручність
2. Компоненти інтерфейсу користувач-комп'ютер
3. Стратегії розробки інтерфейсу користувач-комп'ютер

Конспект лекції

2.1 Інтерфейс користувач-комп'ютер та його зручність

Раніше при розробці комп'ютерних систем основна увага приділялась точності роботи цих систем. Точність означає, що при надходженні на вхід системи заданих значень на її виході отримують задані результати. Значно менше уваги зверталось на зручність роботи користувача з системою [1,2].

Користувач автоматизованої обчислювальної системи має право очікувати не тільки точних результатів обробки, але і зручності в використанні системи. Тіло людини (користувача) - це механізм, який працює в рамках визначених обмежень і допусків. Очам людини необхідно, щоб образи мали визначений розмір, рівень яскравості, контрастності і розміщувались на зручній відстані. Деякі кольори сприймаються краще інших. Необхідно відмітити і обмеження мозку людини. У людини велика довготривала пам'ять і дуже обмежена короточасна пам'ять. Подібно буферам обчислювальної системи ця пам'ять може легко перевантажуватись. Люди можуть розширювати цю пам'ять за допомогою записів на листках паперу, застосовуючи нові моделі роботи або пристосовуючись до нового способу роботи. Однак така адаптація може приводити до стресів, а як результат до неприйняттого рівня помилок.

Критерій зручності став визначальним в останні роки в зв'язку з широким застосування персональних комп'ютерів з діалоговими режимами роботи в різних галузях народного господарства, побуті, що привело до зростання кількості користувачів неспеціалістів в області інформатики та обчислювальної техніки. Згідно з класифікацією фірми IBM [2] розрізняють такі типи користувачів:

- системний програміст;
- програміст;
- кінцевий користувач.

Кінцеві користувачі - це люди, які хочуть більш ефективно і економно вирішувати свої завдання, використовуючи комп'ютерні засоби. Частіше всього вони не є спеціалістами в області інформатики. Досвід показує, що ці користувачі можуть ефективно вирішувати на машині свої завдання лише в умовах наявності активної допомоги зі сторони системи на всіх етапах вирішення завдання і спрощення методів взаємодії з обчислювальною системою [3].

Зрозуміло, що сучасні обчислювальні системи повинні орієнтуватись на таких користувачів в першу чергу, оскільки більшість користувачів персональних комп'ютерів є кінцевими користувачами. Використовуючи в подальшому термін користувач, ми маємо на увазі кінцевих користувачів.

Таким чином, з точки зору користувача діалогової системи система є зручною для користувача, якщо інтелектуальні зусилля користувача, необхідні для розуміння дій системи і реакції на них мінімальні. Можливо сформулювати ряд вимог до системи з точки зору зручності:

1. Поведінка системи по відношенню до користувача повинна бути гнучкою, тобто користувач не повинен діяти строго визначеним способом.
2. Система повинна вміти розрізняти користувача і пристосовуватись до нього.
3. Поведінка системи повинна бути зрозумілою користувачеві.
4. Система завжди повинна бути готова допомогти користувачеві.
5. Для використання системи не потрібні спеціальні навички та додаткове навчання.
6. Не потрібно зловживати здатністю людини до навчання під час роботи з системою.
7. Система повинна реагувати на порушення взаємодії з користувачем, обумовлені властивостями людини і приймати запобіжні заходи проти цих порушень.

Інтерфейс користувач-комп'ютер включає всі ті аспекти обчислювальної системи, з якими безпосередньо взаємодіє користувач.

2.2 Компоненти інтерфейсу користувач-комп'ютер

Основою будь-якої діалогової системи є зручний інтерфейс "користувач-комп'ютер". Він забезпечує зв'язок між користувачем і процесом, який виконує деяке завдання. З точки зору програмного забезпечення в склад інтерфейсу входять дві компоненти (рис. 1.1):

- набір процесів введення-виведення;
- процес діалогу.

Процеси введення-виведення служать для того, щоб прийняти від користувача або передати йому дані через різні фізичні пристрої. З кожним пристроєм зв'язаний свій процес введення-виведення, задача якого прийняти дані від відповідного пристрою і перетворити їх в внутрішній формат з яким можуть працювати "процеси діалогу". Наприклад, прийняти від клавіатури скан-код клавіші "5" (00000110) і перетворити його в двійкове представлення символу в коді ASCII (00110101). Якщо користувач вирішив скористатись мовним введенням, то слово "п'ять" також буде представлене аналогічним чином. Типовими пристроями введення-виведення сучасних комп'ютерів є такі:

Пристрої виведення

- оперативна текстова і графічна інформація: монохромні і кольорові дисплеї на базі ЕПТ (електронно-променева трубка);
- тверда копія: алфавітно-цифрові друкувальні пристрої (виведення тексту); лазерні принтери (текстовий і графічний вивід); графопобудовувачі (графічний вивід);
- звукове виведення: синтезатори мови (мовне виведення); звукогенератори ("музикальне виведення").

Пристрої введення

- введення довільних даних користувачами: клавіатура (текстове введення); планшети (графічне введення);
- автоматичне збирання інформації: пристрої зчитування документів, сканери;
- позиціонування і вибір: світлове перо, сенсорний екран, маніпулятори типу "миша", "джойстик", шар;
- пристрої мовного введення і машинного зору.
- Для більшості застосувань розробка фізичного інтерфейсу включає в себе вибір необхідних пристроїв і створення на їх основі автоматизованих робочих місць. При виборі пристроїв враховуються такі фактори:
- зміст і формат даних, що обробляються;
- об'єми введення-виведення;
- обмеження, які вводяться користувачем і робочим середовищем;
- обмеження, пов'язані з іншими апаратними і програмними засобами, які використовуються в системі.

Процеси діалогу - це механізм обміну інформацією, який можна розглядати як оболонку, яка включає всі процеси, які входять в систему для виконання визначеного завдання.

Діалог між людиною і комп'ютером можна визначити як обмін інформацією між обчислювальною системою і користувачем, який виконується за допомогою інтерактивного терміналу за визначеними правилами.

Діалог користувач-ЕОМ (електронно-обчислювальна машина) можливо розглядати з двох точок зору. У відповідності з першою ("технічною") для забезпечення діалогу достатньо організувати лише швидкий обмін повідомленнями в режимі безпосередньої взаємодії з ЕОМ. При цьому час очікування суб'єктом чергової відповіді ЕОМ не повинен порушувати процес його мислення. Друга точка зору - під діалогом розуміють спілкування між користувачем і ЕОМ на природній мові. Ця точка зору характерна для розробників систем штучного інтелекту [1].

Метою даної роботи є вивчення особливостей розробки структур діалогу, які базуються на першому підході, оскільки поки що обчислювальна потужність ЕОМ недостатня для реалізації і широкого застосування діалогу на природній мові.

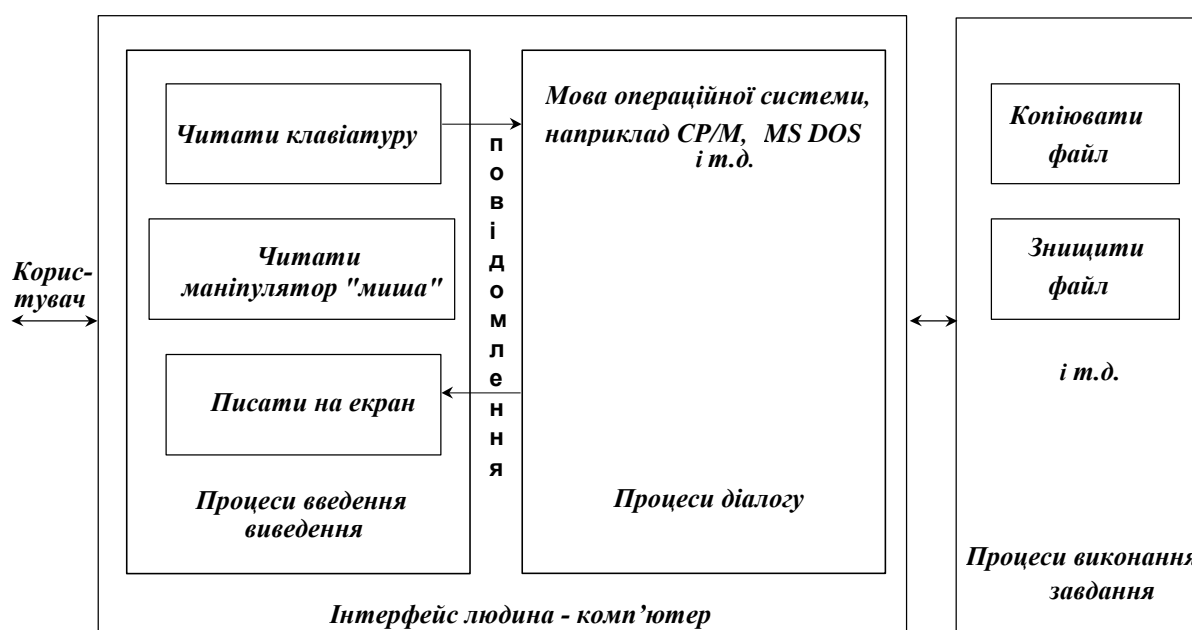


Рис. 2.1. Складові частини інтерфейсу людина-комп'ютер на прикладі операційної системи

Задачі процесів діалогу такі:

- визначити завдання, яке користувач покладає на систему;
- прийняти логічно-зв'язані вхідні дані від користувача і розмістити їх в змінних відповідного процесу в необхідному форматі;
- виклик процесу виконання необхідного завдання;
- виведення результатів обробки по закінченню процесу в придатному (зручному) для користувача форматі.

Незалежно від граматики і способу ведення діалогу в його основі лежить такий цикл:

- явний або неявний запит на введення даних;
- введення даних через процес введення;
- перевірка вхідних даних.

Всі початкові дані, незалежно від формату в якому вони вводяться представляють результат явного або неявного вибору з набору прийнятних відповідей. Якщо діапазон вхідних даних невеликий, то перевірка виконується шляхом порівняння введеного користувачем повідомлення з допустимими іменами завдань. Якщо діапазон вхідних значень великий, то перевірку можна виконати шляхом порівняння форматів вхідного повідомлення з заданим. Наприклад, в запиті

Введіть дату (дд-мм-гг) :

виводиться підказка, яка подає формат введення дати. В такому випадку перевірка виконується шляхом порівняння вхідного повідомлення з заданим форматом.

З точки зору користувача інтерфейс людина-комп'ютер - це основний процес, а процеси, які виконують завдання є неосновними (фоновими).

2.3 Стратегії розробки інтерфейсу користувач-комп'ютер

При розробці інтерфейсу користувач-комп'ютер рекомендується придержуватись наступних правил (стратегій) [1,2]:

- розглядати інтерфейс людина-комп'ютер як окремий компонент системи;
- враховувати можливості апаратних і програмних засобів;
- послідовності, тобто бажано щоб в рамках хоча б однієї операційної системи всі програми мали б однаковий інтерфейс користувача;
- використовувати загальноприйняті в ергономіці рекомендації по взаємодії з машиною взагалі, хоча необхідно враховувати особливості представлення інформації на екрані в різних обчислювальних системах;
- розробник інтерфейсу повинен досконало знати задачу користувача. Він повинен розуміти не тільки обчислювальний процес, необхідний для вирішення задачі, але і оцінити дії користувача, направлені на досягнення мети задачі;
- необхідно залучати користувачів для практичної перевірки розроблених зразків інтерфейсу людина-комп'ютер з наступною доробкою до тих пір, поки не буде створено прийнятний продукт;
- передбачати засоби адаптації в рамках інтерфейсу, тобто він повинен налагоджуватись на потреби різних користувачів або на потреби одного користувача в різні періоди його роботи.

Підсумок

ЗНАТИ: загальні компоненти людино-комп'ютерного інтерфейсу.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Що таке точність обчислювальної системи?
2. Як можливо визначити "зручність"?
3. Що таке інтерфейс користувач-комп'ютер?
4. Компоненти інтерфейсу користувач-комп'ютер?
5. Поясніть призначення процесів введення-виведення.
6. Назвіть типові пристрої введення-виведення.
7. Визначити задачі процесів діалогу.
8. Назвіть основні стратегії розробки інтерфейсу користувач-комп'ютер.

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

Інтерфейс; процеси діалогу; процеси введення-виведення.

Лекція №3. . Класифікація повідомлень в діалозі. Загальні поняття про критерії оцінки якості діалогу

План:

1. Класифікація повідомлень в діалозі
2. Критерії оцінки якості діалогу

Конспект лекції

3.1 Класифікація повідомлень в діалозі

Повідомлення, якими обмінюються учасники під час діалогу виконують декілька функцій, необхідних для підтримки діалогу. Класифікація повідомлень, якими обмінюються людина і комп'ютер під час діалогу приведена на рис. 3.1. [1,4]



Рис. 3.1. Класифікація повідомлень у відповідності з функціями, які вони виконують

Підказка це вихідне повідомлення системи, яке спонукає користувача вводити дані. Наприклад, підказка "ім'я пристрою", зрозуміла для більшості мікрокомп'ютерних систем:

C:_ ,

де С - ім'я дискового пристрою.

Повідомлення про помилку - це сигнал діалогового процесу про те, що подальше виконання роботи неможливе, оскільки він або викликаний процес виконання завдання не може опрацювати повідомлення, введені користувачем.

Вихідні дані - це дані, які повертає процес по закінченню обробки.

Повідомлення про стан системи - це інформація для користувача про те, що відбулося або що відбувається в системі.

Довідкова інформація виводиться в тих випадках, коли користувач не може відповісти на запит системи, тому що не розуміє сам запит або забув що необхідно вводити.

Вхідне керуюче повідомлення або команда вибирає процес для виконання певного завдання, а вхідні дані дають інформацію цьому процесу з якими структурами даних виконувати це завдання.

Діалог можна класифікувати з урахуванням формату вхідного повідомлення і гнучкості, яка дозволяє користувачеві вводити вхідні повідомлення, коли йому зручно. Розрізняють:

- діалог, який керується системою;
- діалог який керується користувачем.

Перший тип діалогу характеризується тим, що система жорстко задає, яке завдання можливо вибрати і які дані вводити. Це виконується за допомогою системи підказок користувачеві. Діалог, який керується користувачем - це діалог в якому ініціатива повністю належить користувачеві, тобто він може подати будь-яку команду на виконання необхідного завдання. Відповідальність за вірність введеної комбінації повністю лягає на користувача.

Формат, в відповідності з яким користувач вводить свої повідомлення називається граматиною діалогу. Існує декілька варіантів граматики діалогу:

- коди;
- ланцюжки ключових слів, подібних до програми;
- обмежена природна мова;
- природна мова.

Реалізація діалогу на природній мові пов'язана з найбільшими труднощами. Хоча існує декілька систем з подібною граматиною, практична цінність такого підходу викликає сумнів.

3.2 Критерії оцінки якості діалогу

Основні критерії оцінки якості діалогу такі:

- природність;
- послідовність;
- ненадмірність;
- підтримка користувача;
- гнучкість.

Природний діалог - це такий діалог, який не спонукає користувача суттєво змінювати свої традиційні способи вирішення задач. Як мінімум це означає, що діалог повинен вестись на рідній мові користувача. Розробник повинен не допускати безглузлого порядку постановки питань. Фрази, по можливості, не повинні вимагати додаткових пояснень. Неприродний діалог часто є наслідком того, що розробник системи не знайомий з тими способами, якими вирішує задачу користувач.

Діалог, який відрізняється логічною послідовністю, гарантує, що користувач, який добре володіє роботою однієї частини системи, легко зможе оволодіти роботою інших частин даної системи. Стандартні відповіді повинні бути дійсно "стандартними". Наприклад, довідка отримується по натисканню однієї клавіші F1 в будь-якій точці діалогу, ALT X - завершення роботи будь-якої програми в даному операційному середовищі. Послідовність в використанні форматів даних означає, що аналогічні поля завжди будуть представлятися системою в одному і тому ж форматі. Наприклад, якщо поле для роботи з датою має формат дд/мм/рр, то всі аналогічні поля повинні мати такий же формат. Послідовність у розміщенні даних на екрані в різних ситуаціях гарантує, що користувачеві відомо, де на екрані шукати інструкції, повідомлення про помилки і т.д.

Ненадмірний (короткий) діалог вимагає від користувача введення тільки мінімуму інформації, необхідної для роботи системи. Нема необхідності в діалозі запитувати інформацію, яку можливо сформулювати автоматично або яка була вже введена раніше,

наприклад поточна дата. Вихідні повідомлення повинні містити тільки ту інформацію, яку запитує користувач.

Підтримка користувача в процесі діалогу - це міра допомоги, яку діалог надає користувачеві під час його роботи з системою.

Основні складові цієї підтримки такі:

- кількість і якість інструкцій, які є в системі;
- характер повідомлень про помилки;
- підтвердження будь-яких дій системи.

Гнучкість діалогу - це міра того, наскільки він відповідає різним рівням підготовки і продуктивності праці користувача. Це передбачає, що діалог повинен вміти адаптувати свою структуру і вхідні дані у відповідності з рівнем підготовки користувача.

Підсумок

ЗНАТИ: класифікацію повідомлень в діалозі та критерії оцінки якості діалогу.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Дайте класифікацію повідомлень в діалозі.
2. В чому різниця між підказкою і довідкою?
3. Поясніть призначення повідомлення про стан системи.
4. Дайте класифікацію діалогу з урахуванням вхідного повідомлення.
5. Дайте характеристики основних граматик діалогу?
6. Приведіть основні критерії оцінки якості діалогу.
7. Як ви розумієте "послідовність діалогу"?
8. Наведіть ознаки, які характеризують "природний" діалог.

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

діалог; діалог, що керується комп'ютером; діалог, що керується користувачем.

Лекція №4. . Структури діалогу. Діалог типу запитання-відповідь та командного типу

План:

1. Структури діалогу
2. Структура діалогу типу запитання-відповідь
3. Структура командного типу діалогу

Конспект лекції

4.1 Структури діалогу

Традиційно виділяється чотири основних структури [1,2]:

- запитання -відповідь;
- меню;
- екранні форми;
- діалог на основі команд.

Кожна з цих структур ґрунтується на аналогії взаємодії між людьми. Ці чотири основні структури діалогу можна розглядати як різновидності структури запитання - відповідь. Їх можливо представити таким чином:

- меню : опереджувальна довідка;
- форма : опереджувальне запитання;
- команда : опереджувальне введення.

Не дивлячись на те, що більшість систем мають в своїй основі структури діалогу типу запитання-відповідь, меню або структуру на основі команд, рідко вдається побудувати діалог для всієї системи, використовуючи тільки одну структуру. Для різних частин діалогу необхідні різні структури в залежності від їх конкретних характеристик, тобто більшість діалогів ґрунтується на змішаних структурах.

4.2 Структура діалогу типу запитання-відповідь

Структура діалогу типу запитання-відповідь(Q&A) оснований на аналогії з звичайним інтерв'ю. Цей тип діалогу відноситься до діалогу, який керується комп'ютером. В кожній точці діалогу система виводить в якості підказки одне запитання, на яке користувач дає одну відповідь. В залежності від отриманої відповіді система може вирішити, яке наступне запитання задавати. Якщо відповідь невірна система виводить повідомлення про помилку і знову виводить підказку. Цей процес повторюється до тих пір, поки не буде отримано прийнятну відповідь. Структура Q&A не гарантує мінімального об'єму введення, який оцінюється в кількості натискань клавіш. Однак, при відповідному підборі скорочень, надлишковість може бути зменшена. Для прискорення введення системи з структурою Q&A повинні мати механізм випереджального введення відповідей. Хоча структура Q&A з появою потужних засобів відображення інформації дещо застаріла, вона має ряд переваг. Це такі:

- структура Q&A є проміжною структурою, яка може задовольнити вимоги різних користувачів і типів даних;
- вона достатньо гнучка і в достатній мірі забезпечує підтримку користувача.

Ця структура застосовується найбільш часто, коли на кожне запитання передбачена велика кількість відповідей, кожна з яких впливає на зміст наступного запитання. Тому структуру Q&A знайшла широке використання в експертних системах.

4.3 Структура командного типу діалогу

Структура діалогу на основі командної мови відноситься до діалогу, що керується користувачем. Команда включає в себе постановку задачі, можливі пояснення і дані, необхідні для її вирішення. Діалог на базі команд працює в телетайпному режимі. Система нічого не виводить крім постійної підказки, Наприклад, підказка "ім'я пристрою", прийнята для більшості мікрокомп'ютерних ОС. Вона означає готовність системи до роботи. Кожну команду вводять з нового рядка і закінчують натисканням клавіші "ENTER". Наприклад,

C:\ DIR <ENTER>

Відповідальність за правильність команд лягає на користувача. Формат команди в загальному випадку приведений на рис. 4.1.



Рис. 4.1. Формат команди

Параметри в списку можна задавати в одному з двох форматів [1]:

- в вигляді позиційних параметрів;
- в вигляді ключових параметрів.

Значення параметра, який задається в позиційній формі, визначається за його місцем в командному рядку. Наприклад, в команді

COPY thisfile newfile

перший параметр задає файл, який повинен копіюватись, а другий параметр, файл куди необхідно копіювати вміст першого. У випадку ключових параметрів кожне значення параметра позначається ідентифікатором, який задає його призначення. Наприклад, в команді

COPY SOURCE=thisfile DESTINATION=newfile

ключові слова source (джерело) і destination (одержувач) визначають призначення кожного параметра. Недоліком позиційних параметрів є те, що помінявши місцями параметри можна отримати протилежний результат, а недоліком ключових параметрів є необхідність запам'ятовування великої кількості ключових слів.

Список параметрів може містити ключі, які уточнюють спосіб інтерпретації команди. Ключі помічаються спеціальним ідентифікатором (наприклад "-", "/").

Структура діалогу на основі мови команд за своїми можливостями найшвидша і найгнучкіша із всіх структур діалогу, але для роботи в цьому режимі необхідна хороша підготовка користувача.

Підсумок

ЗНАТИ: структури діалогу.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Які основні структури діалогу використовуються в комп'ютерних системах?

2. Охарактеризуйте відомі структури діалогу як різновидність структури типу "запитання-відповідь".
3. Які аналоги структури діалогу типу "запитання-відповідь"?
4. Переваги, недоліки, застосування структури діалогу типу "запитання-відповідь".
5. Поясніть принцип роботи діалогу командного типу.
6. Які основні способи завдання параметрів в діалозі командного типу та їх переваги і недоліки?
7. Проаналізуйте діалог командного типу з точки зору користувача.

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

Діалог запитання-відповідь; командний діалог.

Лекція №5. . Діалог типу меню та екранна форма. Інтелектуальні інтерфейси

План:

1. Структура діалогу типу "меню"
2. Структура діалогу на основі екранних форм
3. Інтелектуальні інтерфейси
4. Підтримка користувача

Конспект лекції

5.1 Структура діалогу типу "меню"

Структура діалогу типу "меню" характеризується тим, що у користувача є список можливих варіантів для введення, із якого

<i>Персонал</i>	<i>Клієнти</i>	<i>Меблі</i>	<i>Пристрої</i>	<i>Різне</i>	<i>Кінець</i>
-----------------	----------------	--------------	-----------------	--------------	---------------

а)

<i>Персонал</i>
<i>Клієнти</i>
<i>Меблі</i>
<i>Пристрої</i>
<i>Різне</i>
<i>Кінець</i>

1. *Персонал*
2. *Клієнти*
3. *Меблі*
4. *Пристрої*
5. *Різне*
6. *Кінець*

в)

б)

Рис. 5.1. Приклади різних форматів "меню". а - меню в вигляді рядка даних; б - меню в вигляді блоку даних; в - меню в вигляді списку об'єктів, які вибираються указанням цифрових кодів

необхідно вибрати ті, що потрібно. Популярність меню особливо виросла в зв'язку з появою інтерфейсу, що керується маніпулятором типу "миша" і появою "віконної" техніки [5,6] виведення даних. Існують різні формати представлення меню на екрані. Деякі з них приведені на рис. 5.1.

Меню в вигляді блоку даних на екрані традиційний формат. Меню в вигляді рядка даних може появлятися вверху або внизу екрана і часто залишається в цій позиції протягом діалогу. У випадку, коли на екрані недостатньо місця, додаткові меню "випадають" або "вискакують" (POP-UP) на екран в поточній позиції екрана або витягуються (PULL-DOWN) з рядка меню. Ці меню зникають після вибору варіанта. Меню у вигляді піктограм - це множина блоків об'єктів вибору, розкиданих по екрану, які часто містять графічне представлення об'єктів роботи. Користувач діалогового меню може вибрати необхідний пункт вводячи текстовий рядок, який ідентифікує цей пункт; указуючи на нього безпосередньо, наприклад за допомогою маніпулятора типу "миша", або продиляючи список в режимі ролика і вибираючи з нього. Основні переваги і недоліки меню такі:

1. Меню використовується для введення значень даних, діапазон яких невеликий і їх можна згрупувати в ієрархію.
2. Меню природна структура при роботі з пристроями указання і вибору (наприклад маніпулятор типу "миша").
3. Функціональні клавіші можна налаштувати так, щоб кожна з них відповідала пункту меню.
4. Просто сформувати повідомлення про помилки, оскільки прийнятні варіанти відповідей відображені на екрані.
5. Меню можуть використовувати користувачі будь-якої кваліфікації.

5.2 Структура діалогу на основі екранних форм

Структура діалогу типу "екранна форма" характеризується тим, що система отримує від користувача всю інформацію зразу способом заповнення форми. Аналогом екранної форми є звичайна анкета. На практиці форми використовуються там, де облік якої-небудь діяльності вимагає введення достатньо стандартного набору даних. Користувач може редагувати відповіді, пропускати, повертатись назад. Комп'ютерна система може перевіряти кожне введенне повідомлення зразу після введення або після заповнення всієї форми. Деякі характеристики цієї структури:

1. Працює швидше, ніж Q&A.
 2. В порівнянні з "меню" може маніпулювати більш широким набором вхідних даних.
 3. В меншій мірі підходить для роботи в режимі вибору варіантів, оскільки має послідовну, а не деревоподібну організацію.
 4. Знаходить застосування в базах даних.
- Може слугувати користувачам будь-якої кваліфікації.

5.3 Інтелектуальні інтерфейси

Всі інтерфейси, розглянуті вище були "сліпі". Пересилка інформації між фізичними процесами введення-виведення і користувачем включала просте механічне перетворення: натискання клавіші X генерувало код Y, який перетворювався монітором клавіатури в послідовність одного або більше символічних кодів (рис. 5.2). Процес діалогу діє як пасивний посередник. Йому не потрібні знання про внутрішню структуру робочих процесів, про дані з якими він маніпулює, а необхідні лише деякі правила перетворення форматів. Адаптивні засоби, які розглядаються в п. 3.4, забезпечують лише розширення цих механізмів, які бажано застосовувати користувачеві.

"Інтелектуальні" інтерфейси [1] відрізняються тим, що перетворення, які до цього часу виконувались через механізм визначень, повинні виконуватись в контексті предметної області, що відображається. Такий інтерфейс повинен мати деякі знання про світ задачі в якій функціонує він та користувач (рис. 5.3).

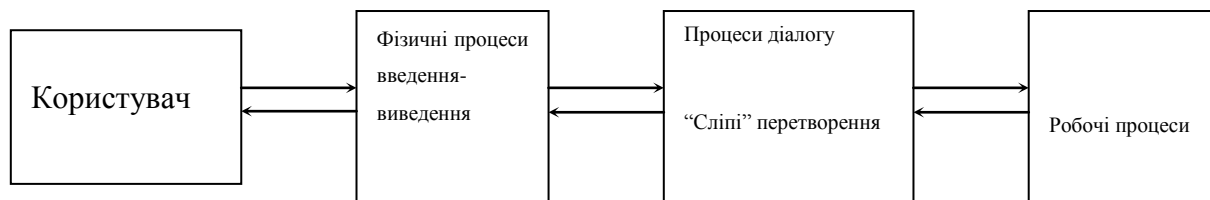


Рис. 5.2. Неінтелектуальний інтерфейс користувача

Необхідно зазначити, що "інтелектуальні" інтерфейси поки що не забезпечують природного діалогу людини з комп'ютером, а лише розширюють взаємодію між людиною і комп'ютером за допомогою:

- збільшення діапазону способів введення і виведення, за допомогою яких виконується взаємодія. Наприклад, за рахунок, мовного введення та виведення;
- збагачення граматики введення-виведення;
- спроби кооперації з користувачем для досягнення мети задачі.

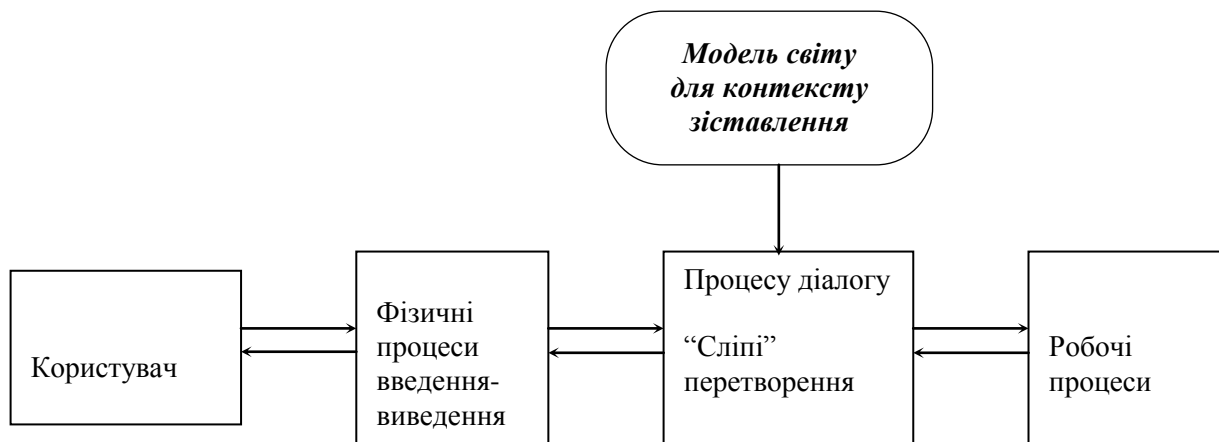


Рис. 5.3 Інтелектуальний інтерфейс користувача

Важливою особливістю інтелектуальних інтерфейсів є те, що вони використовують методи теорії розпізнавання образів для інтерпретації вхідних повідомлень від користувача в світлі системної моделі світу.

Для реалізації інтелектуальних інтерфейсів, необхідний процесор великої потужності і великі об'єми пам'яті. Однак, якщо навіть припустити, що подібний процесор існує, залишаються проблеми представлення образів в системній моделі і удосконалення старих образів.

5.4 Підтримка користувача

Структура діалогу є одним з основних факторів, який впливає на зручності роботи з ним. Однак, для успіху діалогу важливі не тільки рамки в межах яких іде обмін повідомленнями, але і зміст цих повідомлень. Можна виділити такі види підтримки користувача [1,6]:

- підказки;
- повідомлення про помилки;
- довідка;
- керівництво користувача.

Підказки є основою підтримки користувача в діалогах, що керуються комп'ютером. Якщо ця базова підтримка недостатня і користувач допускає помилки при введенні інформації, система повинна виводити повідомлення про помилки, які давали б достатню інформацію про помилку і шляхи її виправлення. Необхідно уникати загрозливих формулювань у повідомленнях про помилки. Крім того, необхідна додаткова підтримка у вигляді довідкової інформації, яка виводиться на екран по запиту користувача, а також надрукована копія "керівництва користувача" (зовнішня документація).

Довідкова інформація і повідомлення про помилки мають багато спільного. Вона повинна бути:

- інформативною;
- своєчасною (доступною в будь-якій точці діалогу);
- відповідати контексту, в якому була викликана.

Щоб відповідати цим вимогам довідкова система повинна мати ієрархічну структуру.

Зовнішня документація в діалогових системах інша, ніж в системах з пакетною обробкою, оскільки користувач не в змозі під час діалогу користуватись об'ємним керівництвом. Достатньо щоб вона містила:

- загальний огляд в межах 10 сторінок;
- опис того, як необхідно почати роботу з системою;
- довідник стандартних засобів, наприклад, в вигляді кишенькового довідника;
- довідкові написи на апаратурі;
- список несправностей.

Підсумок

ЗНАТИ: діалог типу меню та екранної форми, інтелектуальні інтерфейси.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Які аналоги структури діалогу типу "меню"?
2. Які відомі формати представлення "меню" на екрані?
3. Які переваги і недоліки "меню"?
4. Наведіть основні характеристики структури діалогу типу "екранна форма".
5. В яких випадках використовується структура діалогу типу "екранна форма"?
6. Приведіть схему проходження інформації в звичайних інтерфейсах.
7. Приведіть схему "інтелектуального" інтерфейсу.
8. За рахунок чого інтелектуальні інтерфейси розширюють взаємодію з користувачем?
9. Дайте порівняльний аналіз відомих структур діалогу.
10. Які види підтримки користувача застосовуються в діалогових системах?
11. Дайте характеристику довідкової інформації.
12. Дайте характеристику зовнішньої документації.

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Интерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

Діалог типу меню; екранні форми; інтелектуальний інтерфейс.

Лекція №6. Розміщення інформації на екрані. Час відповіді системи

План:

1. Розміщення інформації на екрані
2. Час відповіді системи Інтелектуальні інтерфейси

Конспект лекції

6.1 Розміщення інформації на екрані

На екрані повинна бути присутня лише та інформація, яка необхідна користувачеві в даний момент. Інформація повинна бути організована так, щоб її розташування відображало логічні зв'язки, а формат був зручним для користувача. Для досягнення узгодженості, розміщення даних на всіх екранах повинно відповідати стандартному шаблону екрана (рис. 6.1).

<i>Заголовок і дані про стан системи</i>
<i>Верхня область для виводу повідомлень</i>
<i>Основна область</i>
<i>Нижня область для виводу повідомлень</i>
<i>Опис функціональних клавіш</i>

Рис. 6.1. Шаблон екрана

З незначними змінами цей шаблон був прийнятий за основу при проектуванні панелей (вікон) в CUA (Common User Access) одній з чотирьох компонент стандарту корпорації IBM [5,6], яка визначає правила проектування текстових і графічних інтерфейсів користувача.

Важливе значення для сприйняття інформації має щільність її розташування на екрані. Щільно розташований текст на екрані уповільнює процес його сприйняття. Тому рекомендується дотримуватись таких правил:

- залишати вільною приблизно половину екрана;
- залишати пустий рядок після кожного 5-го рядка таблиці;
- залишати 4-5 пропусків між стовпцями таблиці.

Якщо пов'язану інформацію необхідно розбити на декілька екранів, то необхідно уважно слідкувати за тим, щоб на одному екрані була вся необхідна інформація для вирішення конкретної задачі.

Виділення областей екрана притягує до них увагу користувачів і полегшує орієнтування в інформації розташованій на екрані. Найбільш часто використовується колір для виділення областей екрана. Однак і тут необхідно дотримуватись деяких правил:

- використовувати мінімальну кількість кольорів на одному екрані, не більше трьох або чотирьох;
- використовувати для виділення даних більш яскраві кольори, а більш спокійні відтінки для фону;
- для виділення двох областей - чорний колір або колір з одного кінця спектру використовується для однієї з них, а білий колір або колір з середини спектру для іншої з них;
- необхідне проведення експерименту на реальному екрані.

Оскільки розміщення даних на екрані найбільш часто виконується методом суб'єктивної експертизи, то необхідна автоматизована система розміщення даних на екрані.

6.2 Час відповіді системи

Час відповіді системи - це інтервал між подією і реакцією системи на неї. Що таке час відповіді системи пояснює рис. 6.2.

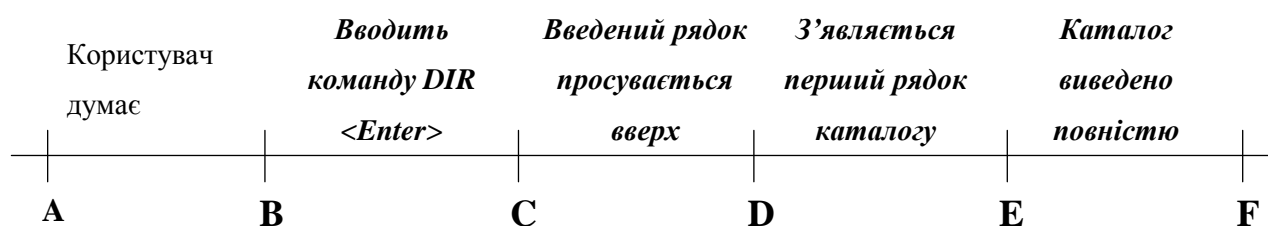


Рис. 6.2. Час відповіді на прикладі команди **Dir** - вивести каталог

Вважається, що час відповіді системи - це інтервал часу CE, тобто час від моменту введення останнього символу запиту до моменту виведення системою першого символу повідомлення, що запитувалось. Інтервал CD - називають "простою відповіддю".

Не існує єдиного прийнятного часу відповіді системи, а лише декілька видів цього часу. Час відповіді можна класифікувати таким чином [1]:

- 0,1-0,2 с.: підтвердження фізичних дій, наприклад натискання будь-якої клавіші;
- 0,5-1 с: відповідь на прості команди;
- 1,0-2,0 с.: зв'язний діалог;
- 2,0-4,0 с.: обробка складних запитів.

Якщо час відповіді більший 15-20 с., то це не діалогова система.

Людина розбиває свою діяльність на етапи, що відповідають порціям інформації, яку вона може зберігати в своїй короткочасній пам'яті. Завершення етапу називається клаузою. Тому затримки перед завершенням етапу неприпустимі, оскільки інформація в короткочасній пам'яті вимагає постійного поновлення і легко стирається. Щоб знизити негативний вплив тривалого часу відповіді, система повинна дозволяти користувачам самим встановлювати клаузи в своїх відповідях за рахунок випереджального введення відповідей та локальної обробки.

Інший спосіб зниження негативного впливу тривалого часу відповіді, це дати користувачеві впевненість, що система все ще працює. Наприклад, виводиться напис "очікуйте", або інформація про стан програми – наприклад, "оброблено 10 % записів". Часто виводяться графічні зображення в вигляді лічильників.

Підсумок

ЗНАТИ: основи розміщення інформації на екрані та класифікацію часу відповіді системи.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Приведіть типовий шаблон екрана.
2. Яка щільність розміщення інформації на екрані?
3. Які правила виділення кольором інформації на екрані?
4. Що таке час відповіді системи?
5. Приведіть класифікацію часу відповіді системи.
6. Як знизити негативний вплив тривалого часу відповіді системи?

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

екран; проста відповідь; час відповіді системи.

Лекція №7. Опис діалогу

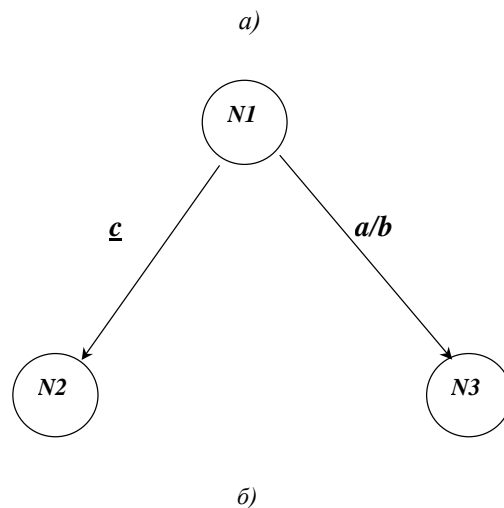
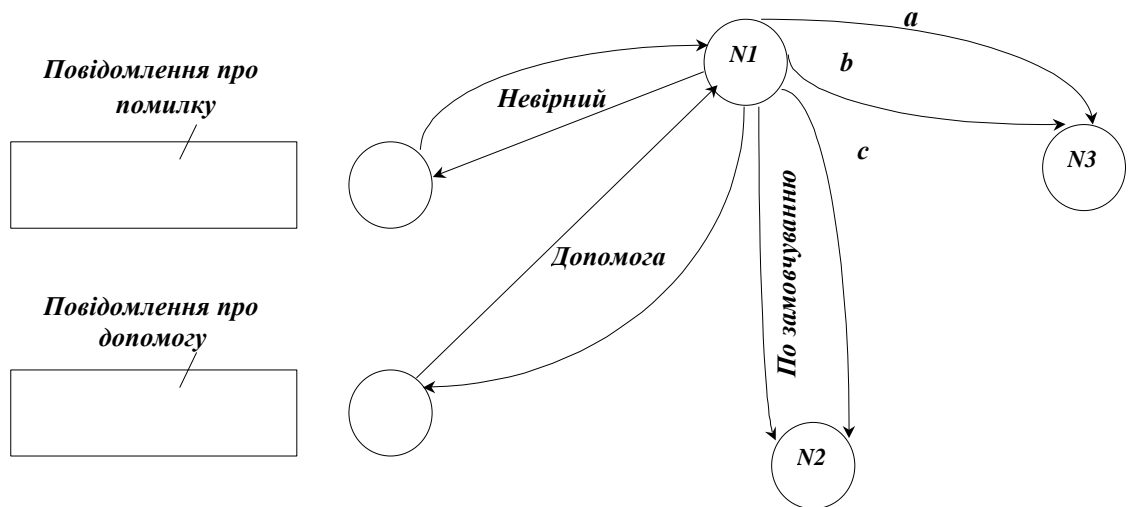
План:

1. Види опису діалогу
2. Мережі переходів
3. Продукційні системи

Конспект лекції

7.1 Види опису діалогу

Для представлення діалогу широко використовуються нарівні з макетами екрана мережі переходів та продукційні системи. Мережі переходів відображають динаміку діалогу, але швидко стають громіздкими, продукційні системи більш компактні, але не показують динаміку діалогу. Ні один з цих двох методів не відображає візуального виведення, тому їх можна вважати доповненням макетів екрана. Макет екрана представляє моментальний знімок діалогу в визначеному стані. Навіть, якщо макети представлені ієрархічно, вони дають погану уяву про те, як діалог переходить з стану в стан.



7.2 Мережі переходів

При опису діалогу за допомогою мережі переходів, кожний стан діалогу представляється пронумерованою вершиною графа(рис. 7.1 а). Вершина - це деяка точка в якій діалог виводить повідомлення користувачеві або вимагає вхідного повідомлення від користувача. Зв'язки між вершинами позначаються направленими дугами. Мітка на дузі визначає умову, при виконанні якої можливий перехід. Можна виділити три типи вершин в діалозі:

- вершина, в якій виводиться повідомлення користувачеві з запитом на введення. Передача на сусідню вершину залежить від контексту введеного повідомлення.
- вершина, в якій виводиться повідомлення користувачеві без запиту на введення, керування автоматично передається на сусідню вершину.
- вершина, в якій виводиться повідомлення користувачеві з запитом на введення, після якого виконується безумовний перехід на сусідню вершину.
- Оскільки мережі переходів швидко стають громіздкими, то їх зображають в згорнутому вигляді (рис. 7.1 б):
- вершини, в яких виводяться допоміжні повідомлення не показуються як окремі вершини;
- якщо пара вершин зв'язується декількома дугами(показує наявність синонімів), то в згорнутому вигляді зв'язок зображується однією дугою і позначається як показано на рисунку.

Кожна вершина в мережі переходів представляє окремий стан в розвитку діалогу і утворює точку перемикання в діалозі.

Перемиканням можна керувати за допомогою структури даних (рис.7.2), яка встановлює відповідність між умовою і наступною вершиною.

Умова	Наступна вершина
C(1)	N(1)
C(2)	N(2)
...	...
C(k)	N(k)

Рис. 7.2. Структура керування маршрутизацією в вершині

Різні типи вершин в мережі переходів характеризуються різними структурами керування маршрутизацією в вершині. Розрізняють два способи організації програмного забезпечення в вершині мережі:

- через підпрограми;
- обробка в вершині визначається через дані - діалог, що налаштовується за таблицею.

Другий спосіб має переваги, оскільки дані змінити простіше.

7.3 Продукційні системи

Продукційні системи зручні для опису синтаксису вхідних повідомлень командної мови. Основою цього підходу є граматичний розбір команди. Граматичний розбір команди - це процес перевірки послідовності знаків на предмет відповідності їх правилам виводу. Наприклад, щоб перевірити, чи є вірною конструкція CHECK A: MYFILE.PAS система будує дерево розбору (рис. 7.3).

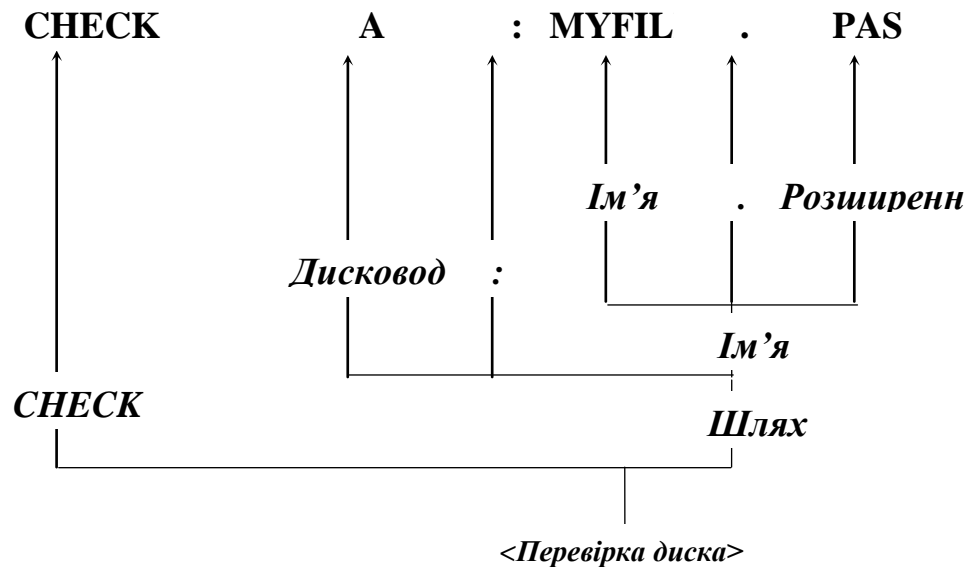


Рис. 7.3. Дерево розбору для рядка CHECK A: MYFILE.PAS

Правила виведення - це компактний спосіб визначення допустимих форматів командної мови, який знаходить також широке застосування для визначення вірних виразів в мовах програмування.

Підсумок

ЗНАТИ: методи опису діалогу.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Які формальні методи застосовуються для опису діалогу?
2. Які недоліки макетів екрану?
3. Які недоліки мереж переходів?
4. Дайте порівняльний аналіз відомих методів опису діалогу.
5. Які способи організації програмного забезпечення в вершині діалогу?

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Интерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

Макет екрану; мережа переходів; продукційна система.

Лекція №8. Опис діалогу з використанням мови UML

План:

1. Основні концепції UML
2. Огляд мови UML

Конспект лекції

8.1 Основні концепції UML

Концептуальна основа об'єктно-орієнтованого аналізу і проектування ПЗ (ООАП) - об'єктна модель.

Основні принципи ООАП:

- абстрагування,
- інкапсуляція,
- модульність,
- ієрархія.

Поняття: об'єкт, клас, атрибут, операція, інтерфейс.

Більшість сучасних методів ООАП засновані на використанні UML.

Уніфікована мова моделювання UML (Unified Modeling Language) - мова для визначення, уявлення, проектування та документування програмних систем, організаційно-економічних систем, технічних систем та інших систем різної природи.

UML містить стандартний набір діаграм і нотацій найрізноманітніших видів. UML - наступник покоління методів ООАП, які з'явилися в кінці 1980-х і початку 1990-х рр.

Створення UML - кінець 1994 р. : Граді Буч і Джеймс Рамбо почали роботу з об'єднання їх методів Booch і OMT (Object Modeling Technique) під егідою компанії Rational Software. Наприкінці 1995 р - створення першої специфікації об'єднаного методу - Unified Method, версія 0.8. У 1995 р до них приєднався творець методу OOSE (Object-Oriented Software Engineering) Івар Якобсон. UML - об'єднання і уніфікація методів Буча, Рамбо і Якобсона, з новими можливостями.

Головні цілі в розробці UML:

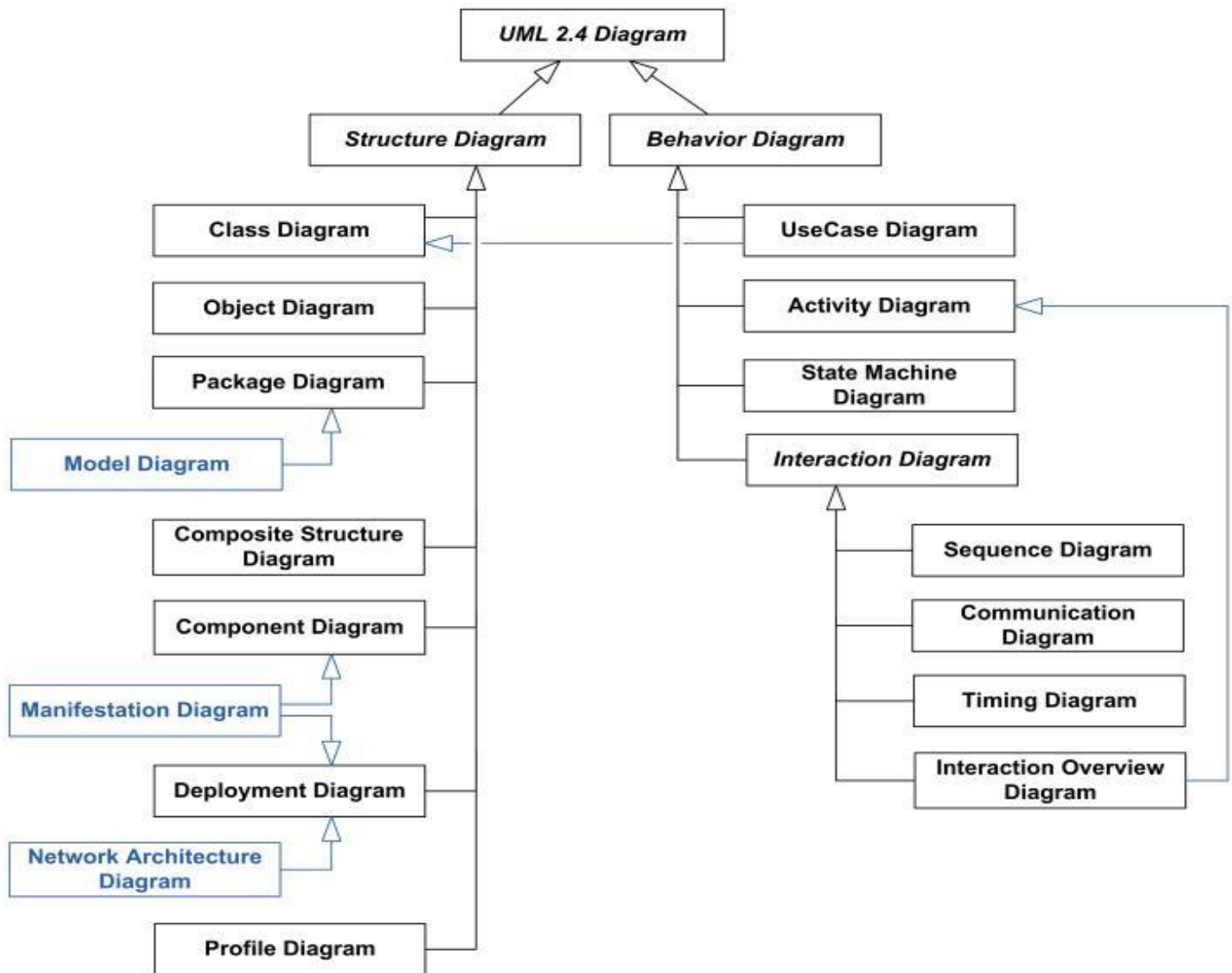
- надати користувачам готовий до використання виразну мову візуального моделювання, що дозволяє їм розробляти осмислені моделі та обмінюватися ними;
- передбачити механізми розширюваності і спеціалізації для розширення базових концепцій;
- забезпечити незалежність від конкретних мов програмування і процесів розробки.
- забезпечити формальну основу для розуміння цієї мови моделювання (мова повинна бути одночасно точним і доступним для розуміння, без зайвого формалізму);
- стимулювати зростання ринку об'єктно-орієнтованих інструментальних засобів;
- інтегрувати кращий практичний досвід.

8.2 Набори діаграм UML

1 Структурні (structural) моделі:

- діаграма класів (class diagram) - для моделювання статичної структури класів системи і зв'язків між ними;

- діаграма об'єктів (object diagram) - показує екземпляри класів і зв'язку між ними;
- діаграма пакетів (package diagram) - показує пакети і зв'язку між пакетами.
- діаграми компонентів (component diagrams) - для моделювання ієрархії компонентів (підсистем) системи;
- діаграма складової структури (composite structure diagram) - показує внутрішню структуру класу і взаємодія елементів внутрішньої структури класу;
- діаграми розміщення (deployment diagrams) - для моделювання фізичної архітектури системи.



2 Моделі поведінки (behavioral):

- діаграми варіантів використання (use case diagrams) - для моделювання функціональних вимог до системи (у вигляді сценаріїв взаємодії користувачів з системою);
- діаграми послідовності (sequence diagrams) і кооперативні діаграми (collaboration diagrams або communication diagrams) - для моделювання процесу обміну повідомленнями між об'єктами;
- діаграма огляду взаємодії (interaction overview diagram) - являє окремий випадок діаграми активності, в якому в якості вершин використовуються інші діаграми;
- діаграма синхронізації (timing diagram) - альтернативне подання діаграми

- послідовності, явно показує зміни стану на лінії життя із заданою шкалою часу;
- діаграми станів (statechart diagrams) - для моделювання поведінки об'єктів системи при переході з одного стану в інший;
 - діаграми діяльності (activity diagrams) - для моделювання поведінки системи в рамках різних варіантів використання (ДБП), або потоків управління.

Підсумок

ЗНАТИ: призначення діаграм UML.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Наведіть основні принципи ООАП.
2. Коли було створено UML?
3. Які головні цілі в розробці UML?
4. Наведіть структурні діаграми UML.
5. Наведіть діаграми поведінки UML.

Додаткові навчальні матеріали

1. <http://www.omg.org>
2. <http://www.rational.com>.

Джерела інформації

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ . [2-е изд.] / Буч Г. - СПб.: Невский Диалект, 1998. - 560 с.
2. Грейди Буч. Язык UML. Руководство пользователя / Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. — СПб.: Питер, 2004. — 432 с.

Глосарій, персоналій

UML; діаграма; use case diagrams, sequence diagrams; statechart diagrams

Лекція №9. Діаграми використання UML

План:

1. Визначення і переваги діаграми варіантів використання
2. Елементи діаграми варіантів використання
3. Приклад діаграми варіантів використання

Конспект лекції

9.1 Визначення і переваги діаграми варіантів використання

Діаграми варіантів використання показують взаємодії між варіантами використання і діючими особами, відображаючи функціональні вимоги до системи з точки зору користувача.

Мета побудови - документування функціональних вимог в загальному вигляді (вимога - простота).

Варіант використання - послідовність дій (транзакцій), виконуваних системою у відповідь на подію, що ініціюється деяким зовнішнім об'єктом (дійовою особою).

Варіант використання описує типове взаємодія між користувачем і системою і відображає уявлення про поведінку системи з точки зору користувача.

Найпростіший випадок: варіант використання визначається в процесі обговорення з користувачем тих функцій, які він хотів би реалізувати, або цілей, які він переслідує по відношенню до розроблюваної системи.

Переваги моделі варіантів використання:

- визначає користувачів і кордони системи;
- визначає системний інтерфейс;
- зручна для спілкування користувачів з розробниками;
- використовується для написання тестів;
- є основою для написання документації користувача;
- добре вписується в будь-які методи проектування (як об'єктно-орієнтовані, так і структурні).

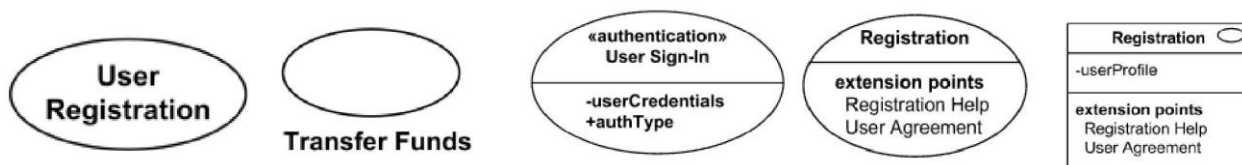
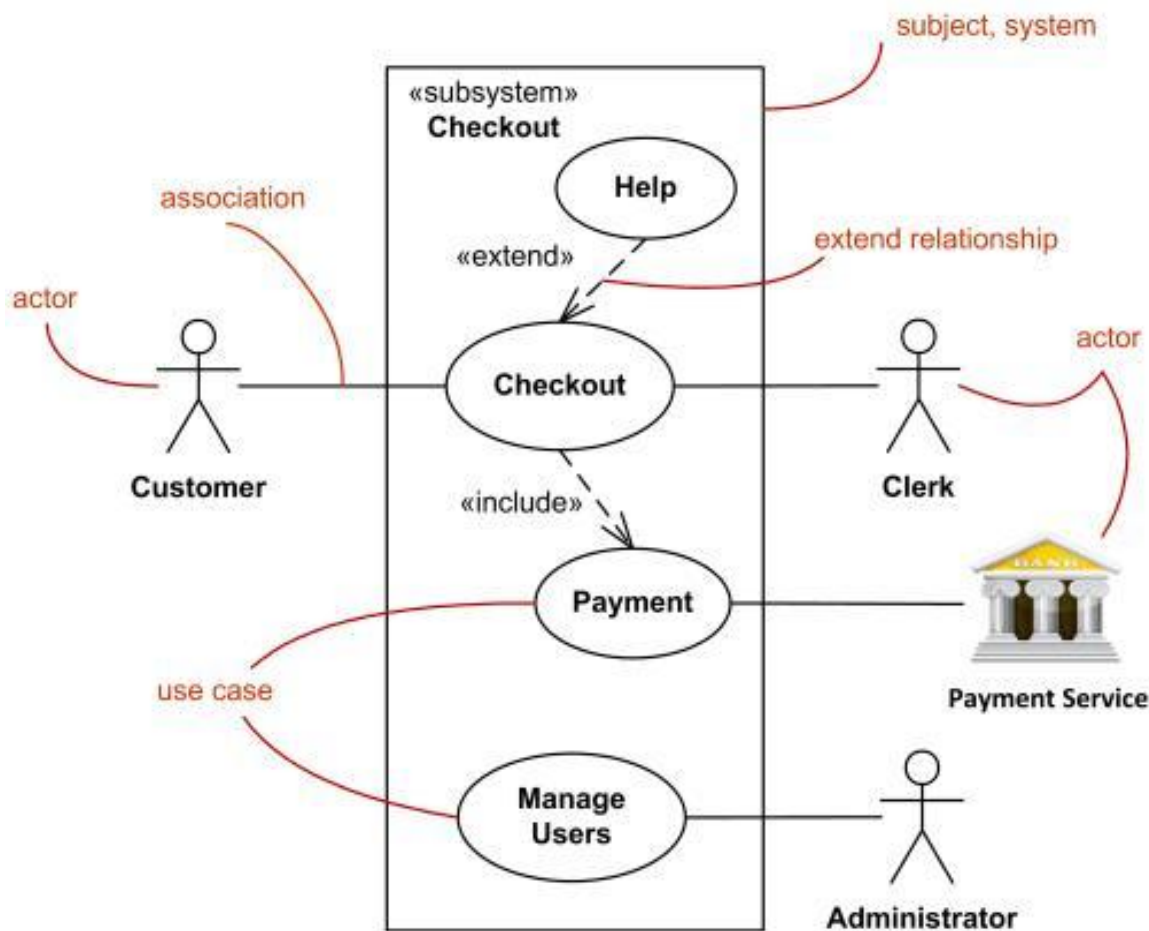
9.2 Елементи діаграми варіантів використання

Варіант використання

Позначається на діаграмі еліпсом, усередині - назва.

Мета варіанту використання - визначити закінчений аспект або фрагмент поведінки деякої сутності без розкриття її внутрішньої структури.

Сутність: система або будь-який елемент моделі, який володіє власною поведінкою.



Актори

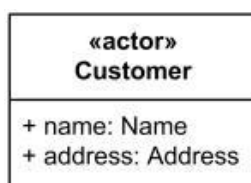
Актор - зовнішня по відношенню до модельованої системи сутність, яка взаємодіє з системою і використовує її функціональні можливості для досягнення певних цілей.

Кожен актор - якась окрема роль щодо конкретного варіанту використання.

Стандартне графічне позначення - фігурка чоловічка, під нею ім'я.

Актори взаємодіють з системою за допомогою обміну повідомленнями з варіантами використання.

Повідомлення - запит актором певного сервісу системи і отримання цього сервісу. Ця взаємодія може бути виражена за допомогою асоціацій між окремими акторами і варіантами використання або класами.



Відносини. Відносини асоціації

Асоціація встановлює, яку конкретну роль грає актор при взаємодії з екземпляром варіанту використання.

Позначення - суцільна лінія між актором і варіантом використання. Лінія може мати умовні позначення:

- ім'я;
- кратність.

Кратність (multiplicity) вказується поряд з позначенням компонента діаграми, який є учасником даної асоціації, і характеризує кількість примірників даного компонента, які можуть виступати в якості елементів даної асоціації.

Позначення - у формі однієї або декількох цифр і символу зірочки.

Поширені форми запису кратності відношення асоціації:

- ціле невід'ємне число (включаючи 0). Вказівка кратності, яка є строго фіксованою для елемента відповідної асоціації. Кількість примірників акторів або варіантів використання одно зазначеному числу;
- два цілих невід'ємних числа, розділені двома крапками. Відповідає нотації для безлічі або інтервалу цілих чисел, яка застосовується в деяких мовах програмування для позначення меж масиву елементів. Безліч цілих невід'ємних чисел, які прямують до послідовно зростаючому порядку;
- два символи, розділені двома крапками. Перший з них - ціле невід'ємне числом або 0, друге - спеціальний символом «*», позначає довільне кінцеве ціле невід'ємне число, значення якого невідомо на момент завдання відповідного відношення асоціації;
- символ «*» - скорочення записи інтервалу «0 .. *».

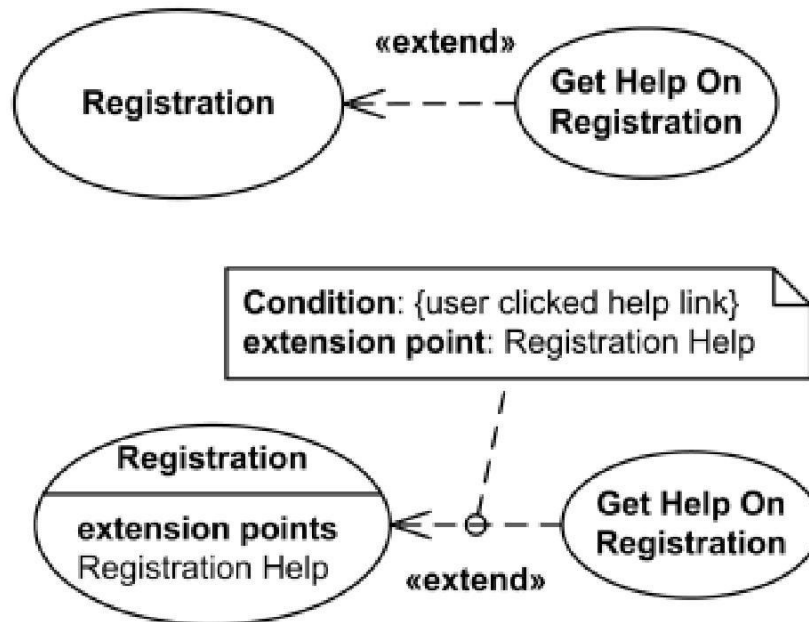
Кратність відносини асоціації за замовчуванням - 1.



Відносини. Відносини розширення

- Відносини розширення - визначає взаємозв'язок екземплярів окремого випадку використання з більш загальним варіантом, властивості якого визначаються на основі способу спільного об'єднання даних екземплярів.
- Позначення - пунктирна лінією зі стрілкою (варіант відношення залежності), спрямованої від того варіанту використання, який є розширенням для початкового варіанту використання. Лінія зі стрілкою позначається ключовим словом «extend» (розширює).
- Один варіант використання може бути розширенням для декількох базових варіантів, а також мати в якості власних розширень кілька інших варіантів.
- Базовий варіант використання може додатково ніяк не залежати від своїх розширень.

Відносини. Відносини розширення



Відносини. Відносини узагальнення

Відносини узагальнення служить для вказівки того факту, що деякий варіант використання А може бути узагальнено до варіанту використання В. В цьому випадку варіант А - спеціалізацією варіанту В. В - предок або батько по відношенню А, а варіант А - нащадок по відношенню до варіанту використання В.

Нащадок успадковує всі властивості і поведінку свого батька, може бути доповнений новими властивостями і особливостями поведінки. Графічне позначення - суцільна лінія зі стрілкою в формі незафарбовані трикутника, яка вказує на батьківський варіант використання.

Застосування: коли необхідно відзначити, що дочірні варіанти використання мають всі атрибути і особливостями поведінки батьківських варіантів. При цьому, дочірні варіанти використання беруть участь у всіх відносинах батьківських варіантів.

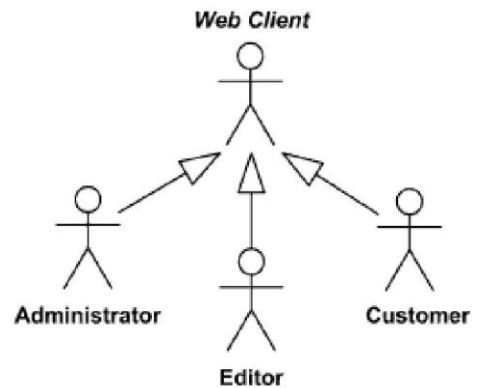
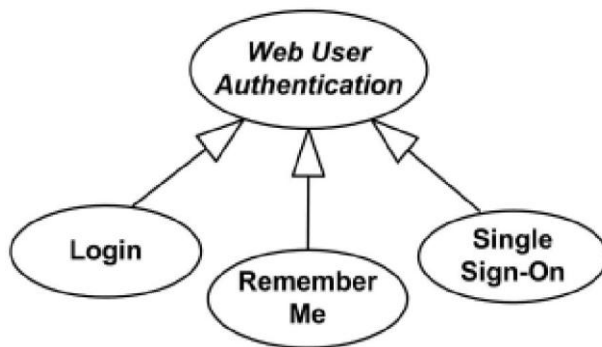
Один варіант використання може мати кілька батьківських варіантів. В цьому випадку реалізується множинне успадкування властивостей і поведінки відносини предків.

Один варіант використання може бути предком для декількох дочірніх варіантів, що відповідає таксономічному характеру відносини узагальнення.

Між окремими акторами також може існувати відношення узагальнення. Дане відношення - спрямоване і вказує на факт спеціалізації одних акторів щодо інших.

Приклад: відносини узагальнення від актора А до актора В відзначає той факт, що кожен екземпляр актора А є одночасно екземпляром актора В і володіє всіма його властивостями. Актор В - батько по відношенню до актора А, а актор А - нащадок актора В. Актор А має здатність грати таку ж безліч ролей, що і актор В. Графічно дане відношення також позначається стрілкою узагальнення.

Відносини. Відносини узагальнення

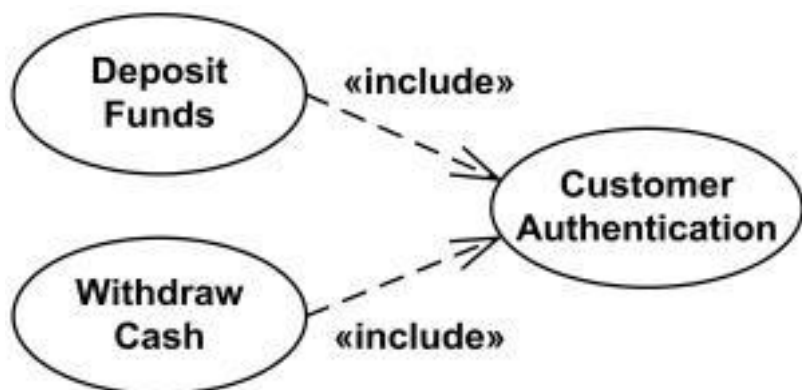


Відносини. Відносини включення

- Відносини включення між двома варіантами використання вказує, що деякий заданий поведінка для одного варіанта використання включається як складовий компонент в послідовність поведінки іншого варіанту використання.
- Один варіант використання може бути включений в кілька інших варіантів, а також включати в себе інші варіанти.
- Включається варіант використання може бути незалежним від базового варіанту в тому сенсі, що він надає йому деякий інкапсульоване поведінку, деталі реалізації якого приховані і можуть бути перерозподілені між декількома включаються варіантами використання.
- Базовий варіант може залежати тільки від результатів виконання включається в нього поведінки, але не від структури включаються в нього варіантів.

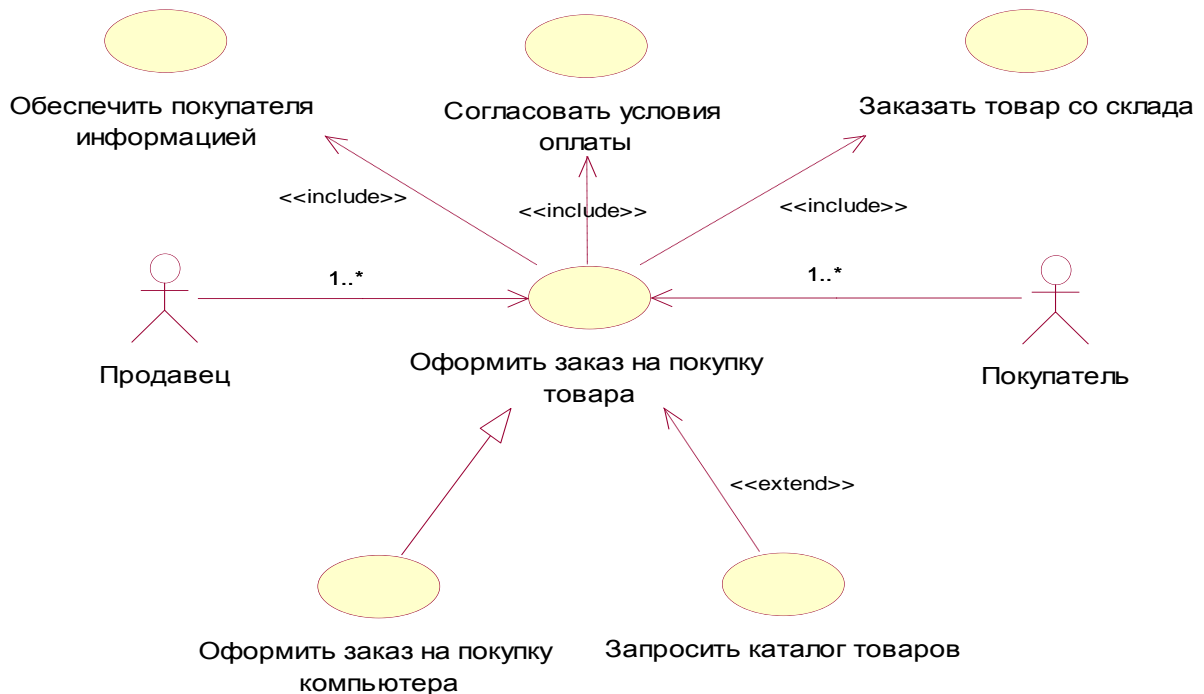
Відносини. Відносини включення

- Відносини включення, спрямоване від варіанту використання А до варіанту використання В, вказує, що кожен екземпляр варіанту А включає в себе функціональні властивості, задані для варіанту В.
- Графічне позначення - пунктирна лінія зі стрілкою, яка позначається ключовим словом «include» (включає).



9.3 Приклад діаграми варіантів використання

Процес оформлення замовлення на покупку товару



Підсумок

ЗНАТИ: елементи діаграм використання UML.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Що таке варіант використання?
2. У які відносини вступають елементи використання?
3. Охарактеризуйте відношення асоціації.
4. Охарактеризуйте відношення розширення.
5. Охарактеризуйте відношення включення.

Додаткові навчальні матеріали

1. <http://www.omg.org>
2. <http://www.rational.com>.

Джерела інформації

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ . [2-е изд.] / Буч Г. - СПб.: Невский Диалект, 1998. - 560 с.
2. Грейди Буч. Язык UML. Руководство пользователя / Грейди Буч, Джеймс Рамбо, Айвар Джекобсон. — СПб.: Питер, 2004. — 432 с.

Глосарій, персоналій

UML; діаграма; use case diagrams; include; extend

Лекція №10. Види адаптації в людино-комп'ютерних інтерфейсах. WIMP-інтерфейс

План:

1. Види адаптації в діалогових системах
2. WIMP-інтерфейс
3. Вікна
4. Переваги і недоліки WIMP-інтерфейсів
5. Критерії проектування WIMP-інтерфейсів

Конспект лекції

10.1 Види адаптації в діалогових системах

Для діалогової системи дуже важливе значення має її здатність адаптуватись користувачем або самостійно до будь-якого можливого рівня підготовки оператора. Існує три види адаптації [1]:

- фіксована;
- повна;
- косметична.

При фіксованій адаптації користувач явно вибирає рівень діалогової підтримки. Очевидно, що новачку потрібно більше засобів підтримки, ніж експерту. Було сформовано правило двох рівнів, відповідно якому система повинна забезпечувати два види діалогу:

- детальний діалог, який забезпечує всебічну підтримку користувача;
- короткий діалог, який призначений для експертів і забезпечує невелику підтримку користувача або зовсім без неї.

Правило двох рівнів було розширене до правила "N" рівнів, при цьому пакет програм дозволяє користувачу явно вибирати один з "N" рівнів діалогу.

Косметична адаптація забезпечує гнучкість діалогових стилів без врахування поведінки користувача і без однозначного вибору ним конкретного стилю діалогу. Це досягається шляхом застосування:

- скорочених і частково збіжних запитів;
- синонімів;
- випереджального введення відповідей;
- відповідей по замовчуванню і макросів;
- багаторівневої допомоги.

Досвідчений користувач може застосовувати їх на свій розсуд. Вони є косметичними в тому сенсі, що як і косметика вони вносять поверхові зміни в базову структуру діалогу, але є при цьому дуже корисними, так як зменшують втому і роблять інтерфейс людина-комп'ютер більш універсальним.

При повній адаптації діалогова система прагне побудувати модель користувача, яка змінюється по мірі роботи останнього з системою і визначає стиль діалогу, адаптуючи його в залежності від цих змін. Дослідження останніх років направлені на те, як в рамках інтерфейсу людина-комп'ютер розробити і підтримувати моделі користувачів на базі методів штучного інтелекту як основи для адаптації.

10.2 WIMP-інтерфейс

Успіхи досягнуті в останні роки в області електронних технологій дозволили значно зменшити собівартість та підвищити потужність комп'ютерних засобів. Їх широке застосування в різних галузях господарства та побуті, привело до значних змін в інтерфейсі людина-комп'ютер. Ці зміни відносяться головним чином до взаємодії з непідготовленими або непостійними користувачами. Типовою серед цих розробок є багатовіконний WIMP-інтерфейс, який можна розшифрувати так:

W - інформація представлена користувачеві на екрані дисплея в виді декількох вікон (windows);

I - об'єкти, з якими система має діло представлені піктографічно в вигляді ікон (icons);

M - вибір об'єктів виконується маніпулятором типу "миша" (mouse);

P - означає меню, яке автоматично впливає (pop-up) на екрані або яке користувач може "витягнути" (pull down) з рядка меню, розміщеного в верхній частині екрана.

WIMP - інтерфейси підтримуються такими операційними системами як Apple Macintosh, Widows, OS/2 та іншими [7,8,9].

В основу WIMP-інтерфейсів покладені три основні метафори [1,5]:

- метафора "конкретний об'єкт";
- метафора "що бачиш - то і одержиш";
- метафора "робочий стіл" та "буфер вивізок".

Метафора "конкретний об'єкт" побудована за аналогією з канцелярським діловодством, де елементи задач представляються конкретними об'єктами. Файли - це пачки паперів, каталоги - висувні ящики в картотеці і т.д. Канцелярські операції передбачають фізичні дії над цими об'єктами. Аналогічно в комп'ютерних системах, наприклад, файл може бути скопійований або стертий "буксировкою" його маніпулятором типу "миша".

Метафора, що "бачиш - то і одержиш" описує такі інтерфейси, в яких фактичний ефект будь-якої дії зразу відображається на екрані дисплея. Наприклад, якщо користувач хоче віддрукувати частину тексту курсивом, то і на екрані вона повинна бути набрана курсивом.

Робочий стіл може містити документи різних форматів, які можуть відноситись до різних задач або відображати різні етапи вирішення однієї задачі (рис. 10.1.).

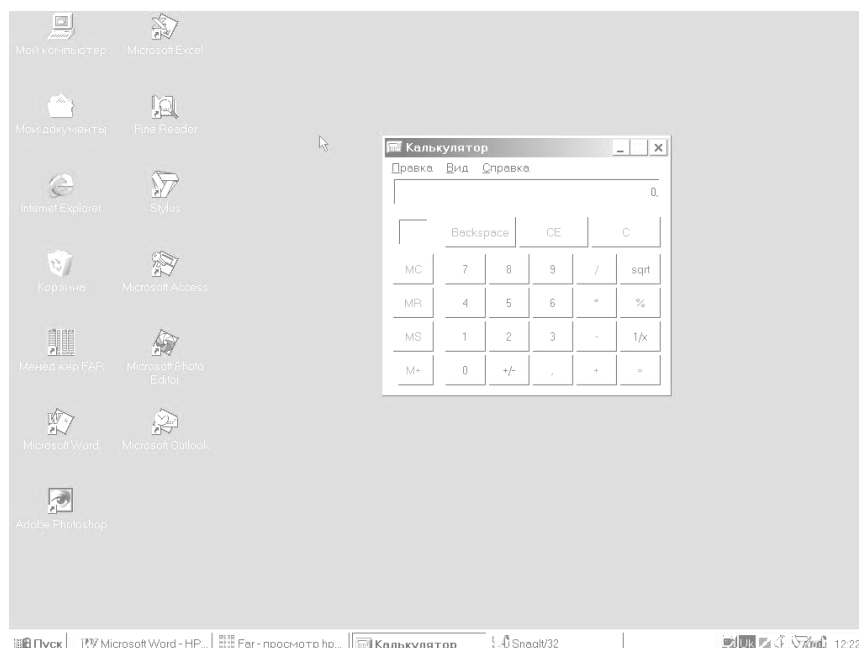


Рис. 10.1. Робочий стіл ОС Windows

Людина може знайти і внести зміни в будь-який з цих документів, перекласти листки з однієї папки в іншу. Аналогічно в комп'ютерних системах метафора "робочий стіл" передбачає, що інтерфейс забезпечує користувачеві доступ до множини різних інформаційних джерел і форматів, дозволяє користувачеві легко переключатись з одного джерела на інше. Також WIMP-інтерфейси забезпечують можливість "вирізки" частини документа з копіюванням її в "буфер вирізок" і наступною вставкою в інші документи стількох копій елементів даних скільки необхідно.

10.3 Вікна

Вікно є спеціальна область (звичайно прямокутної форми) фізичного екрана за допомогою якої користувач оглядає окремі аспекти своєї взаємодії з задачею. На відміну від безвіконного середовища, де вихідні процеси записуються в область пам'яті зарезервовану для відображення фізичного екрана, багатовіконне середовище звичайно вводить проміжний крок як показано на рис. 10.2.



Рис. 10.2. Віртуальні екрани в багатовіконному середовищі

Вихідний процес записує інформацію в допоміжний буфер віртуальний екран. Система може підтримувати декілька допоміжних буферів одночасно. Ці буфери можуть мати будь-який розмір і розташовуватись в будь-якій зручній ділянці пам'яті. Аналогічно як для вихідних процесів є буфери виведення, для вхідних процесів необхідні допоміжні буфери введення, оскільки в персональному комп'ютері є тільки один фізичний буфер клавіатури. Класифікація допоміжних буферів приведена на рис. 10.3.



Рис. 10.3. Класифікація допоміжних буферів

Буфер виведення - це буфер, який діалог заповнює через вихідний процес, користувач не може змінити його вміст. Буфер редагування - це буфер, вміст якого може

змінювати як користувач, так і діалог. Буфер введення - це буфер, в який користувач вводить дані через вхідний процес. В багатовіконному середовищі система повинна зберігати інформацію про відкриті буфери, позицію курсору в кожному буфері та іншу.

Вікно можна розглядати як відображення визначеної порції інформації з деякого буфера на фізичний екран. Кожне вікно має один буфер з яким воно зв'язане, але кожний буфер може мати зв'язок з декількома вікнами. Функції керування вікнами забезпечує система керування відображенням. Вона відповідає за:

- відкриття та закриття вікна;
- переміщення вікна відносно проміжного буфера;
- переміщення вікна відносно екрана;
- зміну розмірів та атрибутів вікна.

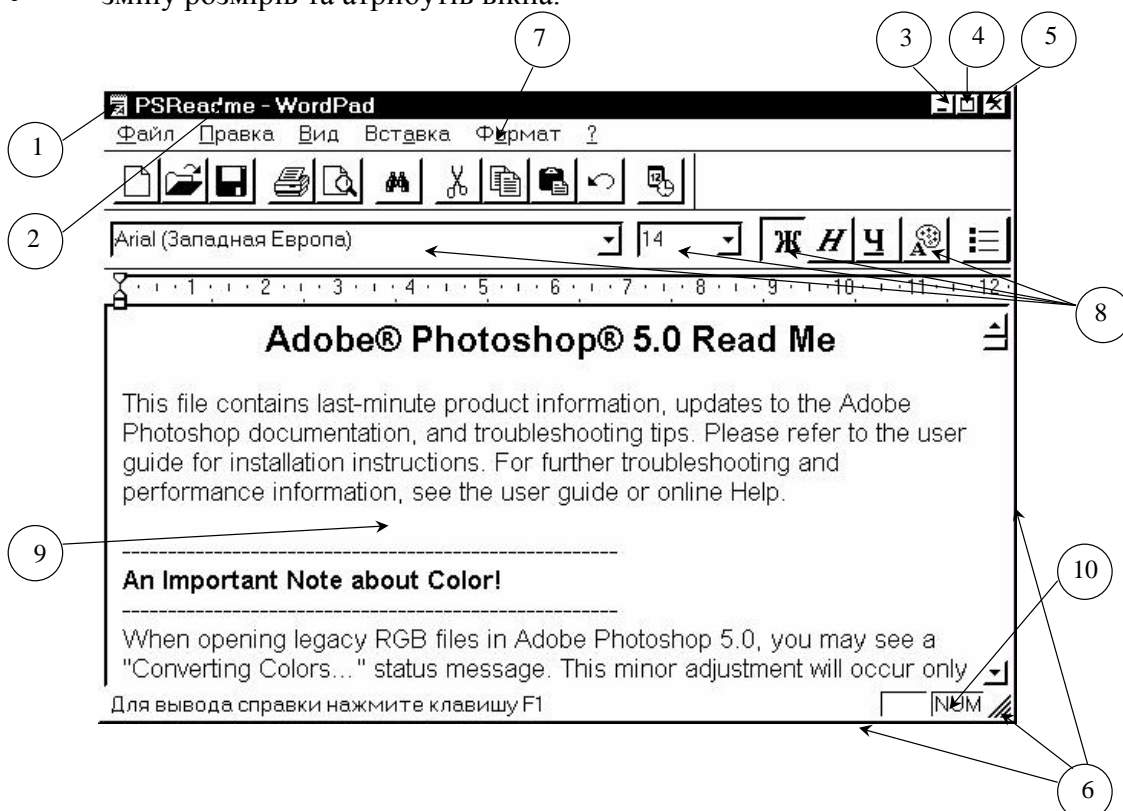


Рис. 10.4. Типове вікно Windows

Різноманітністю вікон є піктограми. Піктограми представляють маленьке вікно з зображенням, яке відображає вміст буфера з яким воно зв'язане. Це вікно може розкриватись для відображення вмісту цього буфера. Піктограми - це корисний засіб нагадування, оскільки вони займають невелику площу і можуть постійно знаходитись на екрані. Типове вікно (або панель - різниця буде пояснена в розділі 5) ОС Windows приведене на рис. 10.4 [7]. Воно містить шість основних елементів:

1. Віконне меню(Control menu): містить декілька команд для налагодження вікна.
2. Область заголовка(Title bar).
3. Кнопка згортання(Minimize button):згортає вікно до кнопки на лінійці задач.
4. Кнопка відновлення / розгортання (Restore/Minimize button): дозволяє збільшити вікно до розмірів екрана або повернути його в стан, який був до розгортання.
5. Кнопка закриття(Close button): закриває вікно.
6. Краї(Edges): за допомогою миші "схвативши" будь-який край вікно можна розтягнути до необхідного розміру.

Внутрішня інформація вікна залежить від його призначення. Але є чотири спільних елементи в середині кожного вікна:

7. Рядок меню (Menu bar): список команд, який розташовується зразу під областю заголовка.
8. Панель інструментів: ряд кнопок, команд, які знаходяться під панеллю меню.
9. Робоча область (Work area).
10. Рядок стану (Status bar): смужка допоміжної інформації.

Крім того вікна можуть мати "лінійки прокрутки", якщо об'єм інформації буфера перевищує розмір вікна, поля введення, кнопки підтвердження дії, відміни і виходу [7,8].

10.4 Переваги і недоліки WIMP-інтерфейсів

Багатовіконна технологія забезпечує користувача готовим доступом до більшого об'єму інформації в порівнянні з одним екраном. Система керування вікнами забезпечує підвищену щільність інформації на екрані. В режимі перекриття вікон користувач може легко переміщувати інформацію з заднього плану на передній. Основні переваги багатовіконної технології такі:

1. Вікна дають доступ до множини джерел інформації. Користувач може об'єднувати інформацію з декількох джерел. Наприклад, зображення одержане за допомогою графічного пакета може бути об'єднане з текстом, одержаним за допомогою пакета підготовки тексту.
2. Споріднену інформацію користувач може досліджувати на різних рівнях деталізації. Наприклад, одне вікно може відображати графік, який характеризує тенденції в торгівлі, а інше цифри, на основі яких побудований графік.
3. Вікна і піктограми дають можливість розширити пам'ять користувача і сприяють ефективному притягненню його уваги. Частково перекрите вікно все ж нагадує користувачеві про свою інформацію; піктограми нагадують про доступні можливості.
4. Пряме маніпулювання звільняє користувача від необхідності вивчати спеціальну комп'ютерну семантику і синтаксис для взаємодії з системою.
5. Інтерфейси прямого маніпулювання характеризуються високим ступенем узгодженості. Велику різноманітність операцій можна виконати невеликим набором механізмів.

Однак, багатовіконні технології мають також ряд недоліків. Основні з них такі:

1. Метафора "робочого стола" може бути аналогією не тільки акуратного робочого стола, але і стола на якому можливий безпорядок. Невпорядкований, дезорганізований стіл може значно скоротити об'єм інформації, який отримує користувач. Це можливо, оскільки концепція вікон дозволяє користувачеві змінювати розміри та позицію вікон.
2. Оскільки користувач може керувати багатовіконним екраном, то існує небезпека, що він буде витрачати більше часу на роботу з дисплеєм, а не з задачами.
3. Піктограми не завжди точно відображають зміст і вимагають великої уваги для розуміння того, що на них зображено.
4. Пряме маніпулювання не обов'язково кращий тип інтерфейсу. Наприклад, групування файлів "резиновою ниткою" є менш швидким в порівнянні з універсальними засобами, які надаються командними мовами.
5. Підтримка багатовіконного WIMP-інтерфейсу, особливо в графічних режимах, вимагає значних затрат ресурсів комп'ютера, що зменшує ресурси, виділені для вирішення прикладних задач.

10.5 Критерії проектування WIMP-інтерфейсів

1. Зміст вікна повинен утворювати логічно-зв'язану групу, а границі кожного вікна повинні бути чітко-позначені.
2. Проектувальник повинен уникати заповнення екрана множиною маленьких вікон, тому вікна повинні появлятися спочатку в відповідних позиціях і мати відповідні

розміри. Система повинна встановлювати по замовчуванню початкову позицію і розмір, з врахуванням вибору користувача.

3. Компонування індивідуального вікна і екрана в цілому повинно виконуватись з урахуванням критеріїв, розглянутих в п. 3.2

4. Розробник повинен уникати складного кодування кнопок "миші", а також складних послідовностей натискання кнопок.

5. Для підготовлених користувачів інтерфейс повинен допускати можливість використання скорочень, командних мов, функціональних ключів як альтернативу керування позицією за допомогою маніпулятора типу "миша".

6. Конфігурація робочої станції повинна мати відповідну потужність для підтримки даного інтерфейсу.

Підсумок

ЗНАТИ: основи побудови WIMP-інтерфейсів.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Дайте порівняльний аналіз основних видів адаптації в діалогових системах.
2. Дайте характеристику фіксованій адаптації.
3. За рахунок чого досягається косметична адаптація?
4. Дайте характеристику повної адаптації.
5. Охарактеризуйте багатовіконні WIMP-інтерфейси.
6. На яких метафорах ґрунтуються WIMP-інтерфейси?
7. Поясніть застосування метафори "робочого столу".
8. Поясніть застосування метафори "що бачиш то і держиш".
9. Поясніть застосування метафори "конкретний об'єкт".
10. Що таке вікна та їх типи?
11. Дайте класифікацію допоміжних буферів WIMP-інтерфейсів.
12. Проаналізуйте переваги і недоліки WIMP-інтерфейсів.
13. Приведіть основні критерії проектування WIMP-інтерфейсів.

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

WIMP; вікно; буфер; екран; карта екрану

Лекція №11. Стандарт фірми IBM по проектуванню інтерфейсів користувача

План:

1. Аналоги і принципи CUA
2. Панелі і вікна в CUA
3. Уніфіковані дії діалогу
4. Допоміжні функції
5. Критерії проектування WIMP-інтерфейсів

Конспект лекції

11.1 Аналоги і принципи CUA

Основною проблемою застосування прикладних систем є проблема їх сумісності та перенесення з однієї системи на іншу. Історично склалося так, що для різних сімейств комп'ютерних систем розроблялись окремі операційні системи і засоби організації інтерфейсів користувача. Однак, широке використання персональних комп'ютерів для керування виробничими системами привело до нових поглядів на проектування цих систем і їх архітектуру. Необхідність обміну програмами, даними, ідеями користувачів різних систем примушує розробляти стандарти і відповідні програмні засоби, які б могли вирішити проблему сумісності і перенесення програм з однієї системи на іншу. Стандарт фірми IBM відомий як Systems Application Architecture (SAA) (архітектура середовища для розробки додатків), багато в чому знімає ці проблеми. SAA містить три компоненти [5,6]:

- система інтерфейсів користувача (Common User Access- CUA);
- система комунікацій (Common Communications Support - CCS);
- система програмних інтерфейсів (Common Programming Interface - CPI)[3,4].

В той час, як CPI забезпечує перенесення прикладних програмних систем, CCS забезпечує концепцію розподільної обробки даних, основною функцією CUA є забезпечення перенесення навиків і знань користувачів в роботі з системами, що не менш важливо.

Система інтерфейсів користувача(CUA) представляє сукупність правил і принципів для керівництва при конструюванні інтерфейсів користувача. В її основу покладені концепції WIMP-інтерфейсів, а саме:

- концепція "робочого стола";
- об'єктна орієнтація;
- указівка і вибір за допомогою маніпулятора типу "миша";
- використання графіки і піктограм;
- багатовіконні технології;
- кероване проектування - керування діями користувача і зменшення інформації, яку необхідно запам'ятовувати.

Проектування діалогу в CUA виконується з урахуванням ряду принципів. Деякі з них такі:

- дії користувача повинні бути оборотними. В межах діалогу дія "відміни" (Cancel) повертає користувача в попередній стан;
- контекст(Context) забезпечує підтримку користувача;

- дії користувача, які є потенційно руйнівними, вимагають підтвердження;
- користувач повинен мати змогу застосовувати "мишу" і клавіатуру в будь-якій точці діалогу за своїм розсудом.

Крім загальних рекомендацій CUA точно визначає ряд параметрів. Так в CUA визначені формати панелей та описані їх елементи. Ідеї закладені в CUA стали основою інтерфейсів користувача багатьох відомих операційних систем. Розробка інтерфейсу згідно з CUA складається з проектування панелей, діалогу та вікон.

11.2 Панелі і вікна в CUA

В CUA визначено ряд термінів необхідних при проектуванні інтерфейсу користувача на основі багатовіконних технологій. Це такі як "ЕКРАН", "ВІКНО", "ПАНЕЛЬ", "РАДІОКНОПКА" та інші.

ЕКРАН - це поверхня дисплея, на якій розташовується інформація призначена для користувача.

ПАНЕЛЬ - це інформація, згрупована і розташована на екрані визначеним способом. Кожну панель можна представити як простір, розділений на три частини, кожна з яких містить окремий вид інформації: меню дій з меню, яке випадає, тіло панелі і область функціональних клавіш (рис.11.1).

Меню дій відображається в верхній частині екрана і надає користувачеві доступ до основних функцій додатку. При виборі дії на екрані з'являється розширення меню дій - випадне меню, яке містить список дій. Деякі панелі можуть не мати меню дій.

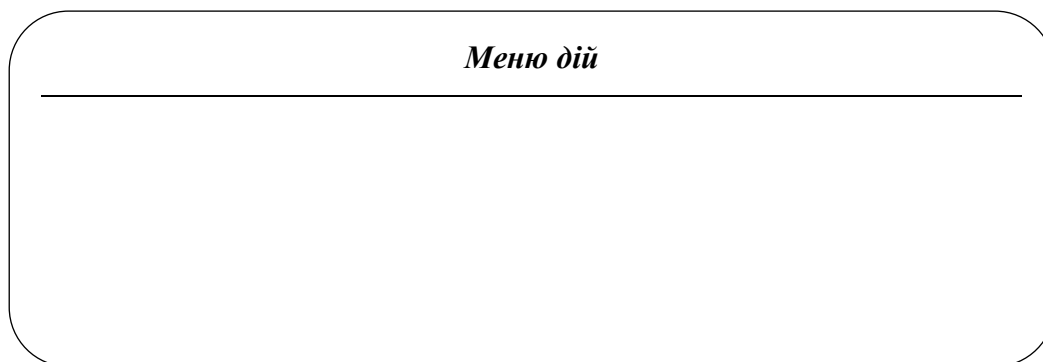


Рис. 11.1. Три області панелі

Тіло панелі може бути розділене на декілька областей. Наприклад, може містити область команд, в якій користувач вибирає команди і область повідомлень, в якій з'являються повідомлення для користувача.

Область функціональних клавіш розташована в нижній частині панелі, користувач може запросити її показ в короткій або довгій формі або виключити. Вона містить список призначень функціональних клавіш.

CUA визначає інтерфейс користувача як для текстових так і графічних додатків. Текстові додатки використовують в інтерфейсі користувача тільки текстову інформацію. Графічні додатки використовують в інтерфейсі користувача графічні елементи: радіокнопка, контактна кнопка, комірка і лінійки прокрутки.

Комірка (Check Box): маленький квадрат за яким іде текст:

☒ CheckBox

Використовується для кожного об'єкта поля багатозначного вибору.

Радіокнопка (Option Button): маленький кружок, за яким іде текст:

☐ OptionButton1

Використовується для кожного об'єкта поля однозначного вибору.

Контактна кнопка: контур, деколи з округленими кутами і з текстом всередині:



Використовується для позначення безпосередніх дій.

Лінійка прокрутки: лінійка з протилежно направленими стрілками



на кінцях і повзунком на ній. Використовується, якщо об'єм інформації перевищує вміст вікна, для прокручування інформації в режимі скролінгу.

CUA визначає п'ять типів панелей:

- меню;
- введення;
- інформаційна;
- панель списку;
- ідентифікації додатку.

Можливо використовувати частини цих панелей для створення комбінованих панелей.

Панель меню містить один або більше списків об'єктів, з яких користувач вибирає один або декілька. Для цього панель містить поля вибору.

Панель введення відображає поля в які користувач вводить інформацію і з яких вибирає об'єкти. Є три типи панелей введення:

- панель введення параметрів відображає поля для введення значень параметрів і для вказівки режимів дій. Містить поля введення і вибору;
- панель введення з заголовками (подібно формі на папері) представляє форму документів на папері.
- панель введення з табуляцією містить поля з заголовками організовані по стовпчиках і рядках.

Інформаційна панель відображає захищену інформацію: дані, повідомлення, довідки.

Панель списку відображає список об'єктів, з яких користувач вибирає один або декілька і запитує одну або декілька дій над ними. Для кожного об'єкта одночасно може запитуватись тільки одна дія, але для різних об'єктів - різні дії.

Панель ідентифікації додатку відображає інформацію, що ідентифікує додаток і авторські права розробників.

ВІКНО - це обмежена частина екрана, в якій розташовані панелі. В CUA визначені три типи вікон: первинні, вторинні і спливальні.

Первинне вікно - це вікно, в якому починається діалог. Кожне первинне вікно може містити стільки панелей, скільки необхідно для ведення діалогу.

Вторинні вікна викликаються з первинних вікон. Це такі вікна, в яких користувач і комп'ютер ведуть діалог паралельно діалогу в первинному вікні. Наприклад, в текстовому редакторі вторинне вікно може містити панель, за допомогою якої змінюється формат документа. Вторинні вікна використовуються також для виведення довідки, яка відноситься до діалогу в первинному вікні. Користувач може переключатись з первинних вікон на вторинні і навпаки.

Спливальні вікна призначені для розширення діалогу користувача з первинним, або вторинним вікном. Одне з призначень спливальних вікон - передача різних повідомлень. Перед тим, як завершити діалог з вікном, користувач повинен завершити роботу з зв'язаним з ним спливальним вікном.

10.3. Уніфіковані дії діалогу

Уніфіковані дії діалогу CUA визначає як дії діалогу, які в усіх додатках мають однакове значення. Це такі:

- відміна;
- команда;
- введення;
- вихід;
- підказ;
- регенерація;
- витягнення;
- ідентифікатори;
- клавіші;
- довідка;
- загальна довідка;
- зміст;
- вказівник;
- опис клавіш;
- навчальна довідка.

Відміна (Cancel) це дія, яка дозволяє користувачеві вернутись на одну панель назад або з випадного меню в меню дій. Відміна повинна включатись :

- в усі випадні меню;
- в панелі, які входять в спливальні вікна, за виключенням інформаційних панелей, які містять інструкцію типу: "Нажміть введення для продовження".
- Область введення команд може бути розташована на основній панелі, у вторинному та спливальному вікні. Правила використання команди такі:
- якщо область команд розміщена у вторинному або спливальному вікні, то вона появляється, коли користувач запитує уніфіковану дію діалогу "команда";
- якщо область команд знаходиться в тілі панелі, то для переходу з області команд в тіло панелі і назад використовується "команда";
- коли курсор знаходиться в полі введення команди, користувач набирає команду і запитує "введення".

Дія "введення" повідомляє додаток, що необхідно обробити інформацію панелі і завжди завершується деяким результатом. Реакція на дію "введення " змінюється від ситуації до ситуації.

Запит на вихід найбільш часто приводить до появи "спливального вікна" з такими об'єктами:

- зберегти і вийти;
- вийти без збереження;
- не виходити.

Коли користувач запитує регенерацію, то вміст поточної панелі або відновлюється в початковий стан, або регенерується для відображення поточного стану.

Витягнення запитується для відображення останньої введенної команди.

Коли користувач запитує дію "ідентифікатори", виконується ввімкнення і вимкнення відображення ідентифікаторів панелей.

Коли користувач запитує підказ, появляється спливальне вікно, яке містить інформацію, яка використовується для заповнення поля введення.

Коли користувач запитує дію "клавiші", то змінюється форма представлення області функціональних клавiш і нижньої частини випадних меню, які містять призначення клавiш. Замовчуванням для області функціональних клавiш є довга форма. Коли користувач запитує цю дію, появляється коротка форма.

Коли користувач запитує підказ, появляється спливальне вікно, яке містить інформацію, що використовується для заповнення поля введення (більш детально п. 5.4)

Коли користувач запитує довідку, появляється панель, яка містить довідкову інформацію (більш детально п.5.4).

11.4. Допоміжні функції

Важливими функціями діалогу є довідка, підказ і повідомлення. Ці функції підвищують ефективність використання додатку.

Довідка (рис. 11.2.) призначена для нагадування користувачеві синтаксису або процедури по використанню панелей або пояснення різних повідомлень. Довідка не є засобом навчання користувачів. Довідка повинна бути контекстуальною, тобто інформація, яка виводиться повинна пояснювати поточну дію діалогу. Панелі довідка можуть відноситись до панелей таких типів:

- інформаційні панелі;
- панелі меню;
- спискові панелі або панелі введення.

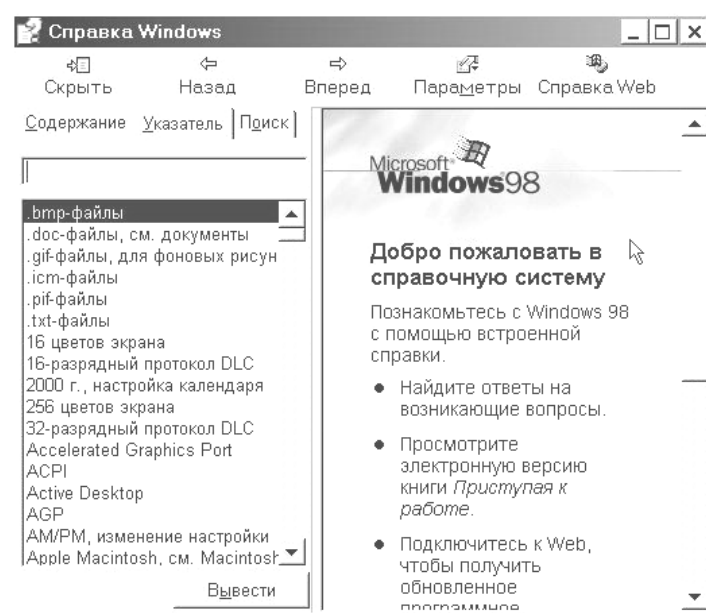


Рис. 11.2 Довідка Windows

Схема розміщення панелі довідка повинна відповідати схемі вибраного типу панелі. Довідка повинна відноситись до об'єкта, на якому в поточний момент встановлено курсор, а не до місця його розташування на панелі.

"Загальна довідка" виводиться, якщо курсор не знаходиться на полі вибору або введення і містить інформацію про панель з якої її запитували.

"Опис клавiш" передбачає панель, яка описує кожен клавiшу додатку і її функції. Цей перелік не повинен вказувати на доступність тієї чи іншої клавiші в поточному стані

додатку. Якщо список достатньо великий і перевищує розмір вікна, то панель повинна мати лінійки прокрутки. Панель відображення клавіш звичайно є інформаційною панеллю.

"Вказівник" передбачає виведення упорядкованого за алфавітом списку тем довідкової інформації додатку. Ця панель представляється панеллю меню з лініями прокрутки при необхідності. Коли користувач вибирає об'єкт з цього меню, з'являється панель "Довідка", яка містить інформацію з вибраної теми.

"Зміст" забезпечує виведення таблиці змісту довідкової інформації, організованої в окремі теми. Зміст може включатись у випадальне меню "Довідка". Для цієї дії в області функціональних клавіш може бути передбачена спеціальна клавіша.

"Навчальна довідка" - це можливість додатку забезпечувати доступ до засобу навчання з панелі, з якою в даний момент працює користувач.

Вся довідкова інформація появляється в одному і тому ж вікні. Коли користувач запитує довідку, вказівник, загальну довідку, опис клавіш, зміст або навчальну довідку, то відповідна інформація заміщує панель, відображену в вікні "Довідка" в даний момент.

Деякі правила створення довідки:

- довідка повинна бути завжди доступною;
- кожна панель додатку повинна мати свою панель "довідка";
- панелі "довідка" повинні містити інформацію про поточну процедуру або про задачу, яку вирішує користувач;
- якщо додаток дозволяє вводити команди, то якщо користувач ввів команду і запросив "довідку", при умові, що курсор знаходиться в області команд, повинна появитись панель, яка описує набрану команду. Якщо користувач запитує довідку без набору команди, то повинна появитись інформація про команди загального характеру;
- уніфіковані дії діалогу і функціональні клавіші повинні діяти таким же чином, як в інших місцях додатку;
- коли користувач входить або виходить з панелі "довідка" дані, введені користувачем до запиту довідка не повинні знищуватись.

"Підказ" підказує користувачам те, як завершити роботу з даним полем введення. Коли запитується "підказ", появляється спливальне "вікно", яке містить панель меню з полем однозначного або багатозначного вибору. Коли користувач вибирає один або більше об'єктів з цього поля, спливальне вікно зникає. Текст вибраного об'єкта копіюється в поле введення. Тобто, завдяки підказу, користувачеві нема необхідності запам'ятовувати і набирати великі об'єми інформації. Достатньо лише вибрати необхідне.

"Повідомлення" забезпечують користувача інформацією, яку він явно не запитував, але яка йому необхідна. Наприклад, повідомлення типу:

"Ви впевнені..."

запитує у користувача підтвердження намірів.

У відповідності з характером інформації, яка виводиться, визначені три типи повідомлень:

1. Інформаційне повідомлення - забезпечує зворотний зв'язок про стан системи. Відповідь користувача не потрібна. Інформаційне повідомлення вказує, що функція виконується або виконалась нормально і використовується в таких випадках:

- користувач багатозадачної системи виконує фонове завдання і воно сповіщає користувача про завершення роботи;
- з'ясування поточного стану системи на основі іншої відображеної інформації може зайняти у користувача декілька секунд;
- користувачеві не ясно, чи обробка продовжується, чи завершилась.

2. "Попереджувальне повідомлення" - звертає увагу користувача на стан, який вимагає деякої дії. Користувач може відповісти на це повідомлення виконанням деякої дії, відмінної від уніфікованої дії діалогу "Відміна". Однак, виконання цієї дії не обов'язково.

3. "Критичні повідомлення" - вказують на умову, коли для продовження роботи користувач повинен виконати конкретну дію. В відповідь на таке повідомлення користувач повинен виконати конкретну дію.

Повідомлення рекомендується виводити в спливальних вікнах. Воно може видаватись в:

- інформаційній панелі;
- панелі меню;
- панелі введення.

Область повідомлення може займати будь-яку кількість рядків. При виводі попереджувальних і критичних повідомлень повинна бути передбачена звукова сигналізація.

Підсумок

ЗНАТИ: елементи графічного інтерфейсу користувача.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. На яких аналогах ґрунтується CUA?
2. Принципи проектування діалогу згідно з CUA.
3. Поясніть такі терміни як "екран", "вікно", "панель".
4. Приведіть стандартний формат панелі CUA.
5. Чим відрізняється інтерфейс користувача текстових і графічних додатків згідно з CUA?
6. Приведіть графічне представлення та призначення радіокнопки, контактної кнопки, комірки, лінійки прокрутки.
7. Які типи панелей визначені в CUA?
8. Які типи вікон визначені в CUA?
9. Які уніфіковані дії діалогу визначені в CUA?
10. Приведіть допоміжні функції діалогу визначені в CUA.
11. Приведіть правила створення довідки згідно з CUA.
12. Які види повідомлень визначені в CUA?
13. Які види довідок визначені в CUA?
14. Які дії діалогу в CUA, коли запитується "підказка"?

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

WIMP; вікно; буфер; екран; карта екрану

Лекція №12. Проектування багатовіконного графічного інтерфейсу в середовищі Microsoft Visual Studio

План:

1. Загальні відомості
2. Приклад створення інтерфейсу графічного додатку

Конспект лекції

12.1 Загальні відомості

Проект Windows Forms Application дозволяє автоматизувати проектування інтерфейсу користувача за рахунок застосування форм Windows подібно візуальним компонентам Delphi [2].

Бібліотека Windows.Forms надає прості й у той же час потужні механізми для керування графічним інтерфейсом користувача.

Форма являє собою екранний об'єкт, який можна застосовувати для надання інформації користувачу і для обробки введення інформації від користувача. Найпростіший спосіб задати інтерфейс користувача для форми - розмістити елементи керування на її поверхні. З іншого боку, форма - це об'єкт, що задається властивостями, що визначають їхній зовнішній вигляд, методами, що визначають їхню поведінку, і подіями, що визначають їхню взаємодію з користувачем. Форми, як і всі візуальні об'єкти в .NET, є екземплярами класів. Зокрема, форми успадковані від System.Windows.Forms.Form.

При проектуванні додатків графічного інтерфейсу користувача програміст взаємодіє зі спеціальним засобом - дизайнером форм. Дизайнер форм призначений для візуальної розробки користувацького інтерфейсу програми. До основних елементів дизайнера форм можна прилічити:

- Properties Window (пункт меню View /Properties Window);
- Layout Toolbar (пункт меню View/Toolbars/Layout);
- Toolbox (пункт меню View/Toolbox).

Проектування графічного інтерфейсу здійснюється аналогічно відповідним технологіям Visual Basic, Delphi, тощо.

12.2 Приклад створення інтерфейсу графічного додатку

Припустимо, нам необхідно створити програму графічного інтерфейсу користувача для роботи з даними про книги на книжковій полиці. Необхідно завантажувати дані з обраного файлу, відображати дані про книги у вигляді таблиці, відображати додаткові дані в окремому підвікні, дозволяти редагування даних, сортувати книги за назвою, знаходити книги з певним автором та дозволяти зберігати змінені дані в новому файлі.

У нашому випадку програма графічного інтерфейсу користувача міститиме таблицю, яка відображає дані про книги, кнопки, пов'язані з конкретними функціями, область відображення даних про авторів.

Після створення нового Windows-додатку на екрані з'являється вікно дизайнера форм. У вікні "Properties" можна налаштувати властивості головного вікна, зокрема властивості Text встановити значення "Книжкова полиця". Далі можна проектувати

інтерфейс користувача шляхом додавання до форми візуальних компонентів, палітра яких міститься у вікні "Toolbox" (View | Toolbox).

Спочатку до форми доцільно додати панель (Panel) для розміщення кнопок. Панель знаходиться на закладці "Containers" вікна "Toolbox". У вікні властивостей (Properties) вставляємо для властивості Dock значення Top. Кнопки (компоненти типу Button) знаходяться на закладці "Common Controls". Напис на кнопці можна змінити, редагуючи властивість text. Між третьою та четвертою кнопками додаємо компонент TextBox - рядок, необхідний для введення прізвища автора, якого ми шукаємо. Цей компонент також знаходиться на закладці "Common Controls".

Для того, щоб кнопки і рядок введення були однакового розміру, були розташовані на однаковій відстані одна від одної, симетрично відносно панелі, слід їх одночасно виділити (наприклад, за допомогою клавіші Shift), а потім застосувати функції підменю Format головного меню: Align | Top, Make Same Size | Both, Horizontal Spacing | Make Equal, Center in Form | Horizontally та Center in Form | Vertically.

Наступний крок - додавання компоненту DataGridView. Це - універсальний компонент для відображення та редагування табличних даних. Він знаходиться на закладці "Data". У першій з доданих таблиць (вона міститься в нижній частині форми) відображатимуться дані про авторів, тому її назву з dataGridView1 (властивість Name у таблиці властивостей) доцільно змінити на dataGridViewAuthors. Для цього компоненту слід налаштувати значення властивості Dock (встановити у Bottom).

Тепер до форми можна додати ще один компонент DataGridView для відображення книжок. Його ім'я слід встановити у dataGridViewBooks. У вікні властивостей встановлюємо для властивості Dock значення Fill.

В результаті форма матиме такий вигляд:

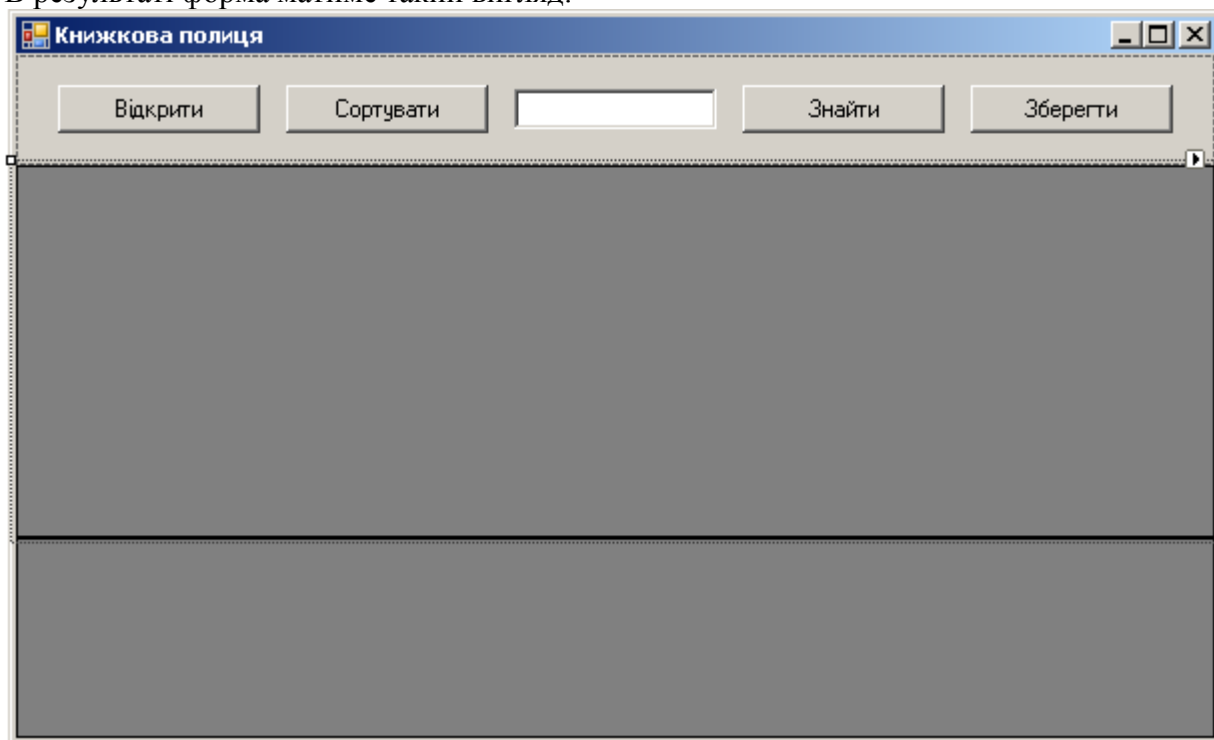


Рисунок 1 - Форма, яка проектується

Для реалізації функції завантаження з файлу та зберігання в іншому файлі до вікна дизайнеру форм слід додати діалогові вікна OpenFileDialog та SaveFileDialog, які знаходяться на закладці Dialogs палітри. Відповідні діалогові компоненти додаються до сірої панелі в нижній частині вікна дизайнеру. Для редагування таблиці (збільшення та зменшення кількості рядків) додаємо контекстне меню (компонент типу ContextMenuStrip). Цей компонент знаходиться на закладці "Menus & Toolbars". Він також додається до панелі в нижній частині вікна дизайнеру. Після подвійного натискання лівої

кнопки миші майбутнє контекстне меню розкривається на головній формі. До нього послідовно додаємо рядки з наступним змістом: "Додати рядок" та "Видалити рядок". Для того, щоб запобігти появі символів кирилиці серед імен змінних, відповідні імена компонентів слід змінити на `addRowToolStripMenuItem` та `removeRowToolStripMenuItem`.

Примітка: для спрощення прикладу ми будемо додавати книги лише з одним автором. Додати можливість зміни кількості авторів можна аналогічно зміні кількості книжок.

Після проектування форми можна перейти до редагування коду. Для переходу на сторінку вихідного коду можна скористатися функцією `View Code` контекстного меню сторінки дизайнера. Частина класу, яка призначена для ручного додавання коду зазвичай має такий вигляд:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
```

```
namespace BookshelfApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

Тепер необхідно реалізувати оброблювачі подій для кнопок. Для додавання оброблювача події треба обрати необхідну кнопку та у вікні "Properties" натиснути кнопку "Events" (з блискавкою). Після подвійного натискання лівої кнопки миші у рядку "Click" можна редагувати код функції-оброблювача, який з'явився у головному вікні. Наприклад, для першої кнопки він спочатку буде таким:

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        // ...
    }
}
```

Всередині тіла функції додаємо код, у якому відображаємо діалогове вікно відкриття файлів. Якщо файл було обрано (результат функції `ShowDialog()` дорівнює `OK`), книжки завантажуються з файлу. У випадку невірної форми файлу (функція `readFromXML()` повернула `false`) відтворюється початковий стан. Функція `MessageBox.Show()` з параметрами, які визначають текст та заголовок вікна, виводить відповідне повідомлення. В обох випадках здійснюється повторна ініціалізація таблиці:

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        if (shelf.readFromXML(openFileDialog1.FileName))
        {
            // ...
        }
    }
}
```

```

    {
        dataGridViewBooks.DataSource = null;
    }
    else
    {
        shelf = new XMLShelf();
        Book book = new Book();
        book.Authors.Add(new Author());
        shelf.addBook(book);
        MessageBox.Show("Помилка читання з файлу", "Помилка");
    }
    initGrid();
}
}

```

Для того, щоб відомості про авторів відображались у другій таблиці, слід додати обробку події RowEnter першої таблиці. Ця подія виникає, коли активізується рядок з певною книгою. Оброблювач цієї події буде таким:

```

private void dataGridViewBooks_RowEnter(object sender,
DataGridViewCellEventArgs e)
{
    int index = e.RowIndex;
    dataGridViewAuthors.DataSource = shelf[index].Authors;
    dataGridViewAuthors.Columns["Name"].DisplayIndex = 0;
    dataGridViewAuthors.Columns["Name"].HeaderText = "Ім'я";
    dataGridViewAuthors.Columns["Surname"].DisplayIndex = 1;
    dataGridViewAuthors.Columns["Surname"].HeaderText = "Прізвище";
    if (shelf[index].Authors.Count == 0)
        shelf[index].Authors.Add(new Author());
}

```

Для того, щоб здійснювати додавання та видалення рядків, буде застосоване контекстне меню. Для того, щоб воно відображалось після натискання правої кнопки миші на основній таблиці, необхідно властивості ContextMenuStrip компоненту dataGridViewBooks встановити значення contextMenuStrip1. Далі для окремих позицій контекстного меню (addRowToolStripMenuItem та removeRowToolStripMenuItem) створюємо функції-оброблювачі подій. До цих функцій додаємо дії, пов'язані з додаванням та видаленням рядків. Отримаємо такий код:

```

private void addRowToolStripMenuItem_Click(object sender, EventArgs e)
{
    dataGridViewBooks.DataSource = null;
    Book book = new Book();
    book.Authors.Add(new Author());
    shelf.addBook(book);
    initGrid();
}

```

```

private void removeRowToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (shelf.Books.Count > 1)
    {
        dataGridViewBooks.DataSource = null;
        shelf.Books.RemoveAt(shelf.Books.Count - 1);
        initGrid();
    }
}

```

```
}  
}
```

Сортування списку з автоматичним відображенням пов'язане з кнопкою button2:

```
private void button2_Click(object sender, EventArgs e)  
{  
    dataGridviewBooks.DataSource = null;  
    shelf.SortByName();  
    initGrid();  
}
```

Для відображення результатів пошуку книг з певним автором можна створити та додати до проекту нову форму (Функція меню Project | Add Windows Form...). До нової форми (Form2) додаємо компонент TextBox. Для цього компоненту слід налаштувати значення властивості Dock (встановити у Fill), а також ReadOnly та Multiline (встановити значення true).

Припустимо, у класі XMLShelf реалізована функція findByAuthor():

```
public List<string> findByAuthor(string surname)  
{  
    List<string> found = new List<string>();  
    foreach (Book book in Books) {  
        foreach (Author author in book.Authors) {  
            if (author.Surname.Equals(surname))  
            {  
                found.Add(book + "");  
                break;  
            }  
        }  
    }  
    return found;  
}
```

У файлі Form2.cs до коду класу Form2 додаємо функцію setContent():

```
public void setContent(string text)  
{  
    textBox1.Text = text;  
}
```

Тепер можна додати оброблювач події button3:

```
private void button3_Click(object sender, EventArgs e)  
{  
    string surname = textBox1.Text;  
    Form2 form2 = new Form2();  
    form2.Text = "Автор: " + surname;  
    List<string> found = shelf.FindByAuthor(surname);  
    string text = "";  
    foreach (string book in found)  
        text += book + Environment.NewLine;  
    form2.setContent(text);  
    form2.ShowDialog();  
}
```

Назва форми Form2 визначається в залежності від того, якого автора ми шукаємо.

Для зберігання даних створюється оброблювач натискання на кнопку button4:

```
private void button4_Click(object sender, EventArgs e)  
{  
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
```

```

    {
        if (!shelf.writeToXML(saveFileDialog1.FileName))
            MessageBox.Show("Помилка запису в файл", "Помилка");
    }
}

```

Тепер проект BookshelfApp можна запустити на виконання.

Підсумок

ЗНАТИ: елементи графічного інтерфейсу користувача.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Концепція програмування, керованого подіями. Поняття події.
2. Для чого потрібні засоби Windows Forms.
3. Як використовують компонент DataGridView.
4. Що таке CheckBox та RadioButton?
5. Де використовується vScrollBar?
6. Для чого використовується компонент TextBox.
7. Назвіть всі відомі Вам елементи керування Windows Forms та їх призначення.

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Глосарій, персоналій

WIMP; вікно; буфер; екран; карта екрану

Лекція №13. WEB-інтерфейс

План:

1. Визначення
2. Варіанти реалізації
3. Переваги застосування
4. WEB-застосунок

Конспект лекції

13.1 Визначення

Веб-інтерфейс — це сукупність засобів, за допомогою яких користувач взаємодіє з веб-сайтом або веб-застосунком через браузер. Веб-інтерфейси отримали широке поширення у зв'язку із зростанням популярності всесвітньої павутини і відповідно повсюдного розповсюдження веб-браузерів.

Однією з основних вимог до веб-інтерфейсів є їх однаковий зовнішній вигляд і однакова функціональність при роботі в різних браузерах.

13.2 Варіанти реалізації

Класичним і найпопулярнішим методом створення веб-інтерфейсів є використання HTML із застосуванням CSS і JavaScript, як правило за допомогою скриптових мов на стороні сервера. Проте різна реалізація HTML, CSS, DOM і інших специфікацій в браузерах викликає проблеми при розробці веб-застосунків і їхньої подальшої підтримки. Крім того, можливість користувача налаштовувати багато параметрів браузера (наприклад, розмір шрифту, кольору, відключення підтримки сценаріїв) може перешкоджати коректній роботі інтерфейсу.

Інший (менш універсальний) підхід полягає у використанні Adobe Flash, Silverlight або Java-апплетів для повної або часткової реалізації користувацького інтерфейсу. Оскільки більшість браузерів підтримує ці технології (як правило, за допомогою плагінів), Flash-або Java-застосунки можуть легко виконуватися. Вони здатні обходити багато несумісності в конфігураціях браузерів, бо надають програмісту більший контроль над інтерфейсом, хоча несумісність між Java або Flash реалізаціями на стороні клієнта може призводити до різних ускладнень.

З розвитком DHTML та JavaScript набув популярності підхід до розробки інтерфейсної частини веб-застосунків, названий AJAX. Серцем технології є здатність веб-сторінки зніціювати запит до веб-сервера і отримати потрібні дані, так щоб інтерфейс не перезавантажував сторінку цілком, а лише довантажують необхідні дані і змінив потрібні частини сторінки, що робить їх більш інтерактивними і продуктивними.

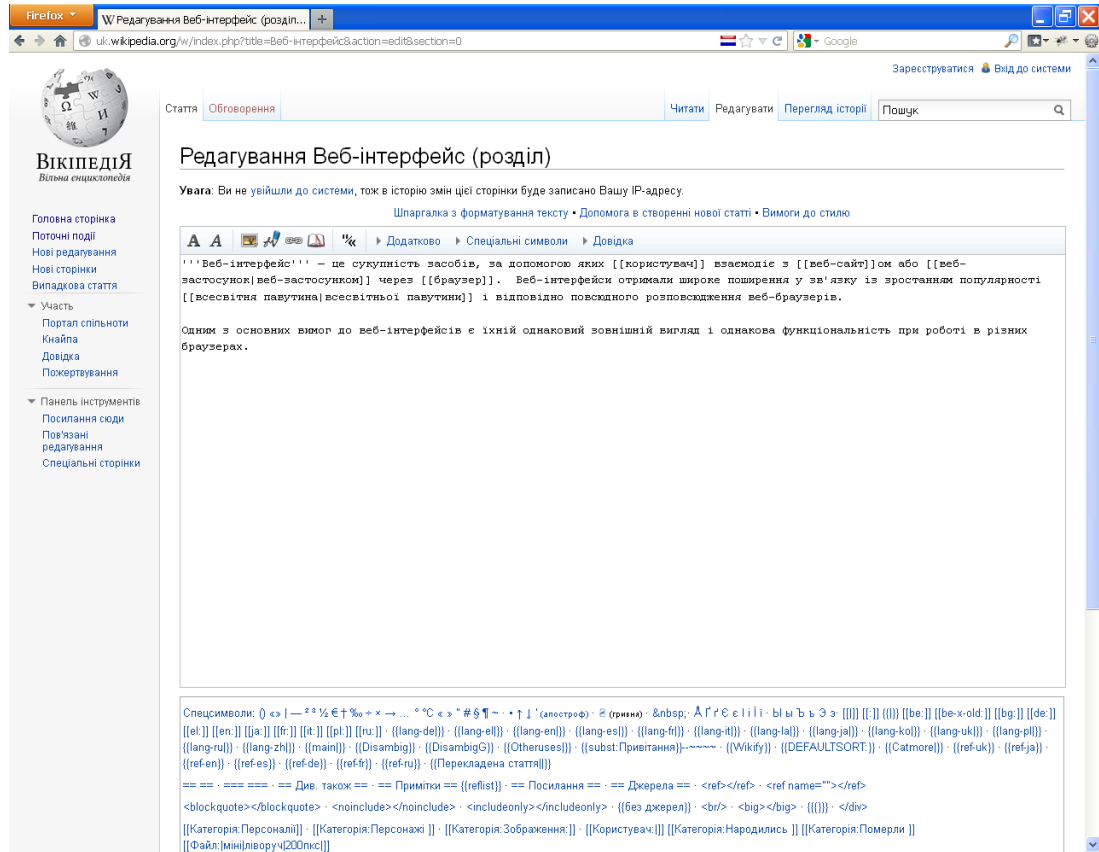
13.3 Переваги застосування

Веб-інтерфейси зручні тим, що дають можливість вести спільну роботу співробітникам, які не перебувають в одному офісі (наприклад, веб-інтерфейси часто використовуються для заповнення різних баз даних або публікації матеріалів в інтернет-ЗМІ).

Добрим прикладом використання і віддачі веб-інтерфейсу є Вікіпедія: практично весь вміст вільної всесвітньої енциклопедії створений і доданий на сторінки сайту за

допомогою веб-інтерфейсу.

Веб-інтерфейс дає можливість універсального віддаленого доступу до служб та пристроїв, у цьому технології практично нема альтернатив. Але водночас, оскільки такий інтерфейс доступний усім, постають серйозні питання забезпечення безпеки, зокрема автентифікація та авторизація користувачів, шифрування переданих даних від сторонніх очей, модерація вмісту тощо.



Приклад веб-інтерфейсу: інтерфейс редагування вікі-тексту вMediaWiki

13.4 Веб-застосунок

Веб-застосунок — розподілений застосунок, в якому клієнтом виступає браузер, а сервером — веб-сервер. Браузер може бути реалізацією так званих тонких клієнтів — логіка застосунку зосереджується на сервері, а функція браузера полягає переважно у відображенні інформації, завантаженої мережею з сервера, і передачі назад даних користувача. Однією з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, тому веб-застосунки є міжплатформовими сервісами. Унаслідок цієї універсальності і відносної простоти розробки веб-застосунки стали широко популярними в кінці 1990-х — початку 2000-х років.

Істотною перевагою побудови веб-застосунків для підтримки стандартних функцій браузера є те, що функції повинні виконуватися незалежно від операційної системи клієнта. Замість того, щоб писати різні версії для Microsoft Windows, Mac OS X, GNU/Linux й інших операційних систем, застосунок створюється один раз для довільно обраної платформи і на ній розгортається. Проте різна реалізація HTML, CSS, DOM й інших специфікацій в браузерах може викликати проблеми при розробці веб-застосунків і подальшої підтримки. Крім того, можливість користувача налаштовувати багато параметрів браузера (наприклад, розмір шрифту, кольори, відключення підтримки сценаріїв) може перешкоджати коректній роботі застосунку.

Інший (менш універсальний) підхід полягає у використанні Adobe Flash або Java-апплетів для повної або часткової реалізації призначеного для користувача інтерфейсу. Оскільки більшість браузерів підтримують ці технології (зазвичай, за допомогою плагінів), Flash- або Java-застосунки можуть легко виконуватись. Оскільки вони надають програмістові більший контроль над інтерфейсом, то здатні обходити багато несумісностей у конфігураціях браузерів, хоча несумісність між Java або Flash реалізаціями клієнта може спричиняти різні ускладнення. У зв'язку з архітектурною схожістю з традиційними клієнт-серверними застосунками, певним чином «товстими» клієнтами, існують суперечки щодо коректності зарахування подібних систем до веб-застосунків; альтернативний термін «Насичений інтернет-застосунок» (англ. *Rich Internet Application*).

Веб-застосунок отримує запит від клієнта і виконує обчислення, після цього формує веб-сторінку і відправляє її клієнтові мережею з використанням протоколу HTTP. Саме веб-застосунок може бути клієнтом інших служб, наприклад, бази даних або стороннього веб-застосунку, розташованого на іншому сервері. Яскравим прикладом веб-застосунку є система управління вмістом статей Вікіпедії: безліч її учасників можуть брати участь у створенні мережевої енциклопедії, використовуючи для цього браузери своїх операційних систем (Microsoft Windows, GNU/Linux або будь-якої іншої операційної системи) без завантаження додаткових виконуваних модулів для роботи з базою даних статей.

Для більшої інтерактивності і продуктивності розроблений новий підхід до розробки веб-застосунків, названий AJAX, і який нині є стандартним де-факто. При використанні Ajax сторінки веб-застосунку здатні відправляти веб-запити до сервера у фоновому режимі, і не перезавантажуються цілком, а лише довантажують необхідні дані з сервера, що значно пришвидшує роботу і робить її зручнішою.

Для створення веб-застосунків використовуються різноманітні серверні технології та мови програмування

Назва	Ліцензія	Веб-сервер
<u>ASP</u>	власницька	спеціалізований
<u>ASP.NET</u>	власницька	спеціалізований
<u>Java</u>	вільна	безліч, зокрема вільних
<u>Groovy</u>	вільна	практично будь-який
<u>Perl</u>	вільна	практично будь-який
<u>PHP</u>	вільна	практично будь-який
<u>Python</u>	вільна	практично будь-який
<u>Ruby</u>	вільна	практично будь-який

Клієнтська частина може використовувати:

- JavaScript
- Flash
- Java / JavaFX
- ActiveX
- Silverlight

Підсумок

ЗНАТИ: особливості web-інтерфейсу.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Дайте визначення WEB-інтерфейсу.
2. Охарактеризуйте варіанти реалізації WEB-інтерфейсу.
3. Яка архітектура WEB-додатку?
4. Охарактеризуйте серверні технології створення WEB-інтерфейсу.
5. Охарактеризуйте клієнтські технології створення WEB-інтерфейсу.

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.
2. Марко Беллиньясо Разработка Web-приложений в среде ASP.NET 2.0: задача — проект — решение = ASP.NET 2.0 Website Programming: Problem — Design — Solution. — М.: «Диалектика», 2007. — С. 640. — ISBN 0-7645-8464-2
3. Олищук Андрей Владимирович Разработка Web-приложений на PHP 5. Профессиональная работа. — М.: «Вильямс», 2006. — С. 352. — ISBN 5-8459-0944-9
4. <https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%96%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81>

Глосарій, персоналій

WEB; інтерфейс; клієнт; сервер

Лекція №14. Інтерфейс Metro

План:

1. Загальні відомості
2. Принципи Metro
3. Шрифт
4. Оцінки

Конспект лекції

14.1 Загальні відомості

Metro - внутрішнє кодову назву художнього стилю оформлення графічного інтерфейсу користувача продукції компанії Microsoft, який використовується з другої половини 2000-х років. Характерні особливості - орієнтація на типографіку і акцент на текст, графічний мінімалізм, плавна анімація, один з відмінних елементів стилю - плоскі прямокутні «плитки».

Спочатку був розроблений для використання в Windows Phone, також використовувався при створенні користувацького інтерфейсу в Microsoft Encarta і MSN 2.0, а також Zune і Windows Media Center. Пізніше на основі Metro був побудований інтерфейс Windows Phone, веб-сайту Microsoft, Xbox 360, Windows 8 [1], Xbox One, Windows 10 (частково).



Знаки в системі перевезень King County Metro, які послужили натхненням для створення Metro

За власною версією дизайнерів Microsoft вважається, що прототипом художнього рішення послужили знаки, часто зустрічаються в системах громадського транспорту - наприклад, розташовані в системі перевезень King County Metro [2], яка обслуговує велику область Сіетла, де розташована штаб-квартира Microsoft.

Ранні елементи стилю Metro зустрічаються в Windows Media Center для Windows XP Media Center Edition, який використовує текст в якості основної форми навігації. Цей інтерфейс був також перенесений в більш пізні версії Media Center. У 2006 році інтерфейс Zune був оновлений з використанням цих принципів. Дизайнери Microsoft вирішили

перепроєктувати інтерфейс з великим упором на чістоту типографіку. Ці принципи і новий призначений для користувача інтерфейс Zune були використані в Windows Phone, багато рішень якої були використані в Windows 8. Плоскі кольорові «живі плитки» були введені в Metro на ранніх стадіях розробки Windows Phone. Microsoft почала інтеграцію цих елементів в інші свої продукти, з прямим впливом, вбачається в більш нових версіях Windows Live Messenger, Live Mesh і Windows 8 [1].

14.2 Принципи Metro

Стиль Metro заснований на принципах дизайну швейцарського стилю. Основними принципами Metro є акцент на гарній типографіці і великий текст, який відразу кидається в очі. Microsoft називає Metro «простим, чистим, сучасним», а також «оновленням» в порівнянні з заснованими на значках інтерфейсами Windows, Android і iOS [3].

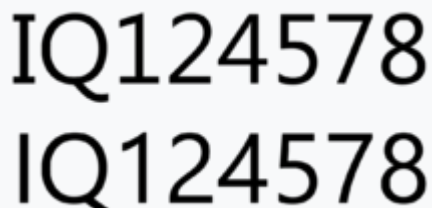
Microsoft розробила Metro спеціально для зміцнення групи загальних завдань для прискорення використання. Це досягається за рахунок виключення зайвої графіки і замість цього опори на фактичний зміст, для функціонування в якості основного призначеного для користувача інтерфейсу.

Велику роль відіграє анімація. Microsoft рекомендує плавні переходи і взаємодія з користувачем на основі реальних рухів (таких, як натискання або переміщення). Це створює у користувача враження «живого» і чуйного інтерфейсу з «доданим відчуттям глибини» [4] [5].

Microsoft склала внутрішній список принципів, що вважаються основою Metro [6].

Приблизно одночасно з офіційним випуском Windows 8 (26 жовтня 2012 року), оскільки все більше розробників і партнерів Microsoft почали працювати над створенням нових додатків в стилі Metro, по цій темі були створені багато веб-сайти з ресурсами, в тому числі особливі правила розробки графічних інтерфейсів для Windows Store [7].

14.3 Шрифт



Нова версія використовуваного в типографіці Metro шрифту Segoe UI (вверху), створена для Windows 8.

- В якості основного шрифту використовуються шрифти з сімейства шрифтів Segoe, розробленого Стівом Меттсон (англ.) з Agfa Monotype і ліцензованого Microsoft. Для Zune Microsoft створила спеціальну версію під назвою Zegoe UI [8], а для Windows Phone - сімейство шрифтів «Segoe WP». Для Windows 8 також була створена спеціальна версія з тією ж назвою Segoe UI.

Всі ці шрифти в основному відрізняються тільки незначними деталями. Найбільш очевидні відмінності між Segoe UI і Segoe WP видно в цифрових символах. Версія Segoe UI, яка використовується в Windows 8, схожа на Segoe WP. Символи з помітними друкарськими змінами в цій версії включають 1, 2, 4, 5, 7, 8, а також I і Q.

14.4 Оцінки

Рання реакція на Metro була в цілому позитивною. В огляді Zune HD, Engadget сказав: «Microsoft продовжує свій ривок до великої, великий типографіке тут, забезпечуючи складний, акуратно розроблений макет, який майже настільки ж функціональний, наскільки привабливий» [9]. CNET похвалив Metro, кажучи, «він трохи більш сміливий і неформальний, ніж жорсткі, стерильні" сітки "з значків і схоже на картотеку меню iPhone і iPod touch» [10].

Суспільство промислових дизайнерів Америки (англ. The Industrial Designers Society of America, IDEA) нагородило Windows Phone, який використовує Metro, золотою нагородою «Народний вибір дизайну», а також премією «Кращий на виставці» [11]:

25 серпня 2012 року Пітер Брайт (англ. Peter Bright) з Ars Technica зробив огляд попередньою версією Windows 8, присвятивши першу частину огляду порівнянню екрана «Пуск» Windows 8 (побудованого за допомогою Metro) і його попередньої версії в Windows 7 і раніше. Перебираючи їх плюси і мінуси, Пітер Брайт прийшов до висновку, що екран «Пуск», хоча і не позбавлений деяких проблем, є явним переможцем [13].

14.5 Відмова від використання назви Metro

У серпні 2012 року почали циркулювати чутки, що один з роздрібних партнерів Microsoft - німецька компанія Metro AG - загрожувала судовим позовом проти Microsoft за порушення прав на товарний знак «Metro». Розробникам і співробітникам Microsoft була розіслана пам'ятка, яка закликала відмовитися від використання терміна «Metro» в зв'язку з результатами переговорів з «важливим європейським партнером» [14] [15]. Microsoft тимчасово називає мову дизайну «Modern UI».

9 серпня 2012 з'ясувалося, що Microsoft планує використовувати термін «Windows 8», щоб замінити «Metro» в споживчих матеріалах, а термін «Modern UI» призначений для розробників, які планують створювати програмне забезпечення на основі нових принципів дизайну [16]. З вересня 2012 року в документації MSDN використовується термін «мова дизайну Microsoft» (англ. Microsoft design language), коли мова йде про мову дизайну [17] [18] [19] [20]. Крім того, на конференції розробників Microsoft Build 2012 компанія також використовує назву «мова дизайну Microsoft» [21] [22].

Для позначення створених в стилі Metro додатків для Windows 8 вживаються терміни «додаток в стилі Windows 8» (англ. Windows 8-style app), і «додаток Магазину Windows» (англ. Windows Store app).



Підсумок

ЗНАТИ: особливості metro-інтерфейсу.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Що таке Metro-інтерфейс?
2. Охарактеризуйте аналоги Metro-інтерфейсу.
3. Охарактеризуйте принципи Metro-інтерфейсу
4. Який тип шрифту використовує Metro-інтерфейс.
5. Чому відмовились використовувати назву Metro?

Додаткові навчальні матеріали

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Джерела інформації

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.
2. MSDN: Общие принципы дизайна приложений в стиле Windows 8
3. Windows Phone Design System — Codename Metro (англ.)
4. Шрифт в интерфейсе Windows 8
5. *Mike Kruzeniski*. From Transportation to Pixels (англ.). Microsoft (17 февраля 2011). — Сообщение в официальном блоге Windows о подобном дизайне. Проверено 21 сентября 2014.
6. [https://ru.wikipedia.org/wiki/Metro_\(%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81\)](https://ru.wikipedia.org/wiki/Metro_(%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81))

Глосарій, персоналій

Metro; інтерфейс; шрифт; windows 8

Лекція №15. Кількісний аналіз інтерфейсу

План:

1. Інформаційна та символна продуктивність інтерфейсу
2. Модель швидкості друку GOMS. Розрахунки за моделлю GOMS
3. Закони Фітса і Хіка

Конспект лекції

15.1 Інформаційна та символна продуктивність інтерфейсу

Якщо мається модель інтерфейсу, то за допомогою GOMS і його розширень можна визначити час, необхідний користувачу на виконання будь-якої, чітко сформульованої задачі, для якої даний інтерфейс передбачений. Однак моделі аналізу не можуть дати відповідь на питання про те, наскільки швидко повинний працювати інтерфейс. Щоб відповісти на нього, ми можемо скористатися мірою, застосовуваної в теорії інформації. Далі ми будемо розглядати термін інформація в технічному змісті, тобто як квантифікацію деякого обсягу даних, переданих за допомогою засобу комунікації, як, наприклад, при розмові двох людей по телефоні, чи якщо людина подає деякий сигнал машині, наприклад за допомогою натискання кнопки ГУВ, коли курсор знаходиться у визначеній області екрана. Перед тим як поглибитися в деталі техніки виміру того, який обсяг інформації потрібний для виконання поставленої задачі, обґрунтуємо необхідність такого виміру.

Щоб зробити правильну оцінку часу, необхідного на виконання задачі за допомогою найшвидшого інтерфейсу, насамперед варто визначити мінімальну кількість інформації, що користувач повинний увести, щоб виконати задачу. Це мінімальна кількість не залежить від моделі інтерфейсу. Якщо методи роботи, використовувані в передбачуваному інтерфейсі, вимагають уведення такої кількості інформації, що перевищує мінімальне, це означає, що користувач робить зайву роботу, і тому інтерфейс можна удосконалити. З іншого боку, якщо від користувача потрібно ввести саму ту кількість інформації, яка необхідна для виконання задачі, то для цієї задачі інтерфейс не можна зробити більш продуктивним шляхом зміни кількості інформації. У цьому випадку шляху поліпшення інтерфейсу (а також багато шляхів для погіршення) усе-таки залишаються, але принаймні дана мета підвищення продуктивності буде вже досягнута.

Інформаційно-теоретична продуктивність визначається так само, як поняття продуктивності визначається в термодинаміку - відношенням потужності на виході до потужності на вході процесу. Якщо протягом якогось періоду часу електрогенератор, що працює від двигуна продуктивністю в 1000 ват, робить 820 ват, то він має продуктивність $820/1000=0.82$. Продуктивність також часто позначається через відсотки. У цьому випадку продуктивність електрогенератора буде складати 82%. Ідеальний генератор (який не може існувати з погляду другого закону термодинаміки) повинний мати продуктивність 100%.

Інформаційна продуктивність інтерфейсу E визначається як відношення мінімальної кількості інформації, необхідної для виконання задачі, до кількості інформації, що повинний увести користувач. Так само як і у відношенні фізичної продуктивності, параметр E може змінюватися в межах від 0 до 1. Якщо ніякої роботи для виконання задачі не потрібно чи робота просто не виконується, то продуктивність складає 1. Продуктивність E може дорівнювати і 0 у випадках, коли користувач повинний ввести інформацію, що зовсім марна. Слід зазначити, що в інтерфейсах можна зустріти чимало

деталей, що мають сумнівну цінність через параметр $E=0$. Прикладом такого марного елемента може бути діалогове вікно, у якому є тільки одна-єдина можливість для дії користувача, наприклад кнопка ОК. У JavaScript є навіть спеціальна команда `Alert`, призначена тільки для того, щоб робити такі непотрібні діалогові вікна. Розроблювачі мови JavaScript були досить розумні, щоб забрати з нього команду `goto` і зробити програмування на цій мові структурним, але вони випустили з уваги аспект інтерфейсу.

У параметрі E враховується тільки інформація, необхідна для задачі, і інформація, що вводиться користувачем. Два чи більш методи дії можуть мати однакову продуктивність E , але мати різний час виконання. Можливо навіть, що один метод має більш високий показник E , але діє повільніше, ніж інший метод, - наприклад `M K M K` і `M K K K`. У цьому прикладі при використанні першого методу повинно бути введено тільки два символи. При використанні другого методу потрібно ввести три символи, але часу на всю дію витрачається менше. Важко привести інші приклади зі звичайного життя, у яких відбувається аналогічна перестановка швидкості й інформаційної продуктивності. Як правило, чим більш продуктивним є інтерфейс, тим більш людиноорієнтованим він є.

Розглянемо різні способи оцінювання кількості інформації подані в. Найбільш простим є комбінаторний (*combinatory*) підхід. Згідно з цим підходом, якщо дискретне джерело інформації U в кожний конкретний момент часу може прийняти випадковим чином один стан із скінченної множини можливих станів N , то ентропія (*entropy*) джерела дорівнює:

$$H(U) = \log N \quad (1.1)$$

Ця міра інформації була запропонована Хартлі в 1928 році. Основа логарифму не має принципового значення і визначає тільки масштаб і одиницю невизначеності. Оскільки сучасна інформаційна техніка базується на двійковій системі числення, то звичайно основу логарифма вибирають рівною двом. При цьому одиниця невизначеності називається двійковою одиницею або бітом (*binary digit - bit*). Якщо основу логарифма вибрати рівною десяти, то невизначеність отримаємо в десяткових одиницях на один стан - дітах (*dit*). Основним недоліком комбінаторного підходу є його орієнтованість на системи з рівномірними станами, хоча реальні системи не є такими.

Імовірнісний (*probabilistic*) підхід враховує цей недолік. В загальному випадку джерело характеризується сукупністю станів (*states*) з імовірностями (*probabilities*) їх появи (ансамблем).

$$U = \begin{bmatrix} u_1 & u_2 & \dots & u_N \\ p_1 & p_2 & \dots & p_N \end{bmatrix} \quad (1.2)$$

$$\sum p_i = 1$$

Для джерела інформації з нерівномірними станами міра невизначеності була запропонована американським вченим К. Шенноном. Згідно зі Шенноном:

$$H(U) = -C \sum_{i=1}^N p_i \log p_i, \quad (1.3)$$

де C – довільне додатне число. Якщо основа логарифма рівна двом, то $C = 1$, і

$$H(U) = -\sum_{i=1}^N p_i \log_2 p_i. \quad (1.4)$$

Таким чином з точки зору теорії інформації $H_{\max}(U)$ визначається мірою Хартлі

$$H_{\max}(U) = \log N.$$

А мінімальна кількість інформації визначається мірою Шеннона

$$H_{\min}(U) = -\sum_{i=1}^N p_i \log_2 p_i.$$

Тоді інформаційна продуктивність інтерфейсу визначається так:

$$E = \frac{H_{\min}(U)}{H_{\max}(U)} \quad (1.5)$$

Для даного аналізу зручніше використовувати більш просту міру, чим теоретична інформаційна продуктивність. Символьна ефективність часто має таку ж практичну цінність, що й інформаційна продуктивність. Вона визначається як мінімальна кількість символів, необхідне для виконання задачі, віднесене до кількості символів, що у даному інтерфейсі потрібно ввести користувачем.

15.2 Модель швидкості друку GOMS. Розрахунки за моделлю GOMS

Метод GOMS – модель, що заснована на оцінці швидкості друку. Розробники, які знайомі з методом GOMS, рідко проводять детальний і формальний аналіз моделі інтерфейсу. Частково це відбувається через те, що основи GOMS і інших кількісних методів відомі їм настільки, що вони спочатку керуються цими методами в процесі розробки. Формальний аналіз використовують у випадках, коли необхідно вибрати один з двох варіантів розробки, коли навіть невеликі відмінності в швидкості можуть давати великий економічний і психологічний ефект. Інколи розробники користуються дуже точні моделі GOMS, як, наприклад, аналіз з використанням методу критичного шляхи GOMS (critical-path method GOMS, CPM-GOMS) або версія, яку називають природною мовою GOMS (natural GOMS language, NGOMSL), в цій моделі враховується поведінка недосвідченого користувача, наприклад час, необхідний для навчання. За допомогою цих методів можна передбачити, скільки часу потрібно користувачеві для виконання деякого набору дій при використанні даного інтерфейсу з абсолютною похибкою менше 5 %. У розширених моделях майже всі оцінки не виходять за межі стандартного відхилення, прийнятого для вимірних значень часу.

Розробники моделі GOMS під час її створення зазначили, що час, потрібний для виконання якогось завдання системою «користувач – комп'ютер», є сумою всіх тимчасових інтервалів, які було потрібно системі на виконання послідовності елементарних жестів, складових дане завдання. Хоча для різних користувачів час виконання того або іншого жесту може сильно відрізнитися, дослідники виявили, що для більшої частини порівняльного аналізу завдань, що включають використання клавіатури і графічного пристрою введення, замість проведення вимірів для кожного окремого користувача можна застосувати набір стандартних інтервалів. За допомогою ретельних лабораторних досліджень був отриманий набір часових інтервалів, потрібних для виконання різних жестів. Нижче наводиться оригінальна номенклатура, в якій кожен інтервал позначений однією буквою (Card, Moran і Newell, 1983).

K=0.2 с	Натискання клавіші. Час, необхідний для натискання клавіші.
P=1.1 с	Вказівка. Час, необхідний користувачу для того, щоб вказати на певну позицію на моніторі.
H=0.4 с	Переміщення. Час, необхідний користувачу для того, щоб перемістити руку з ГПВ на клавіатуру або навпаки.
M=1.35 с	Ментальна підготовка. Час, необхідний користувачу для того, щоб свідомо підготуватись до наступного кроку.
R	Час очікування відповіді комп'ютера.

На практиці вказані значення можуть варіюватися в широких межах. Для користувача, здатного друкувати із швидкістю 135 слів/хв., значення K може складати 0.08 с, для звичайного користувача, що має швидкість 55 слів/хв., – 0.2 с, для середнього недосвідченого користувача, що має швидкість 40 слів/хв., – 0.28 с, а для початківця – 1.2 с. Не можна сказати, що швидкість набору не залежить від того, що саме набирається. Для того, щоб набрати одну букву з групи випадково узятих букв, більшості людей вимагається близько 0.5 с. Якщо ж це якийсь заплутаний код (наприклад, адреса електронної пошти), то у більшості людей швидкість набору складе близько 0.75 символів

в секунду. Значення К включає і той час, який необхідний користувачеві для виправлення відразу відмічених

Обчислення часу, необхідного на виконання тієї або іншої дії (наприклад, «перемістити руку з графічного пристрою введення на клавіатуру і набрати букву»), за допомогою моделі GOMS починаються з перерахування операцій із списку жестів моделі GOMS, які складають цю дію (у наведеному прикладі це Н К). Перерахування рухів (К, Р і Н) – це досить проста частина моделі GOMS. Складнішим, наприклад, в моделі швидкості друку GOMS, є визначення крапок, в яких користувач зупиниться, аби виконати несвідому ментальну операцію, – інтервали ментальної підготовки, які позначаються символом М. Основні правила (по Card, Moran і Newell, 1983, с. 265), що дозволяють визначити, в які моменти проходить ментальні операції, представлені нижче

Правило 0. Початкове встановлення операторів М

Оператори М слід встановлювати перед всіма операторами К (натиснення клавіші), а також перед всіма операторами Р (вказівка за допомогою ГПВ), призначеними для вибору команд; але перед операторами Р, призначеними для вказівки на аргументи цих команд, ставити оператор М не потрібно.

Правило 1. Видалення очікуваних операторів М

Якщо слідуючий після М оператор є повністю очікуваним, з точки зору користувача, то оператор М може бути видалений. Наприклад, якщо ви переміщуєте ГПВ з наміром натискувати його кнопку після досягнення певної області екрану, то відповідно до цього правила, слід видалити оператор М. В цьому випадку послідовність Р М К перетворюється на Р К.

Правило 2. Видалення операторів М всередині когнітивних одиниць

Якщо рядок вигляду М К М К М К належить когнітивній одиниці, то слід видалити всіх оператори М, окрім першого. Когнітивною одиницею є безперервна послідовність символів, що вводяться, які можуть утворювати назву команди або аргумент. Наприклад У, переміщати, Олена Троянська або 4564.23 є прикладами когнітивних одиниць.

Правило 3. Видалення операторів М перед послідовними розділювачами

Якщо оператор К означає зайвий розділювач, що стоїть в кінці когнітивної одиниці (наприклад, розділювач команди, що слідує за розділювачем аргументу цієї команди), то слід видалити оператор М, що стоїть перед ним.

Правило 4. Видалення операторів М, які є переривачами команд

Якщо оператор К є розділювачем, що стоїть після постійного рядка (наприклад, назва команди або будь-яка послідовність символів, яка кожного разу вводиться в незмінному вигляді), то слід видалити оператор М, що стоїть перед ним. (Додавання розділювача стане звичною дією, і тому розділювач стане частиною рядка і не вимагатиме спеціального оператора М.) Але якщо оператор К є розділювачем для рядка аргументів або будь-якого іншого змінного рядка, то оператора М слід залишити перед ним.

Правило 5. Видалення перекриваючих операторів М

Будь-який оператор М, який перекриває оператор Р, що означає затримку, пов'язану з чеканням відповіді комп'ютера, враховувати не слід.

Крім того, відзначимо, що в цих правилах під рядком розумітиметься деяка послідовність символів. Розділювачем вважатиметься символ, яким позначений початок або кінець значимого фрагмента тексту, такого як, наприклад, слово природної мови або телефонний номер. Наприклад, пропуски є розділювачами для більшості слів. Крапка є найбільш поширеним розділювачем, який використовується в кінці речення. Дужки використовуються для обмеження пояснень і зауважень і т. д. Операторами є К, Р і Н. Якщо для виконання команди потрібна додаткова інформація (наприклад, у разі коли для установки будильника користувачеві потрібно вказати час його включення), ця інформація називається аргументом даної команди.

15.3 Закони Фітса і Хіка

Закон Фітса - загальний закон, що стосується сенсорно- моторних процесів, що зв'язує час руху з точністю руху і з відстанню переміщення : чим далі або точніше виконується рух , тим більше корекції необхідно для його виконання , і відповідно , більше часу потрібно для внесення цієї корекції [1] .

Закон опублікований Полом Фітса в 1954 році. Одним з найбільш важливих уроків , який можна зробити з закону Фітса , є висновок , що незважаючи на те , що найлегше клацнути по самій великій кнопці , це не обов'язково буде найважливіша кнопка. Саме через це в безлічі інтерфейсів користувачів кнопки підтвердження просто величезні.

Математично закон записується як

$$T = a + b \log_2 (D/W+1)$$

де :

T - середній час, що витрачається на вчинення дії;

a - середній час запуску / зупинки руху;

b - величина, що залежить від типової швидкості руху;

D - відстань від точки старту до центру цілі;

W - ширина мети , виміряна уздовж осі руху.

Для наближених обчислень a=50, b=150.

Перед тим як перемістити курсор до мети або вчинити будь-яке інше дію з набору безлічі варіантів, користувач повинен вибрати цей об'єкт або дію.

У законі Хіка стверджується, що коли необхідно зробити вибір з n варіантів, час на вибір одного з них буде пропорційним логарифму за основою 2 від числа варіантів плюс 1, за умови, що всі варіанти є рівноімовірними. У цьому виді закон Хіка дуже схожий на закон Фітса :

$$\text{Час (мс)} = a + b \log_2 (n + 1)$$

Якщо ймовірність i-го варіанту дорівнює p(i), то замість логарифмічного коефіцієнта використовується

$$\sum_i p(i) \log_2 ((1/p(i)) + 1)$$

Коефіцієнти , що використовуються у виразі закону Хіка , у великій мірі залежать від багатьох умов, включаючи те, як представлені можливі варіанти, і те , наскільки добре користувач знайомий з системою. (Якщо варіанти представлені незрозумілим чином , значення a і b зростають. Наявність навичок і звичок у використанні системи знижує значення b .) Ми не будемо розглядати ці залежності - для нас важливо , що для прийняття того чи іншого рішення потрібен час ; що для прийняття складних рішень потрібно більше часу , ніж для прийняття простих рішень , і що взаємозв'язок є логарифмічною. При відсутності більш точних даних для проведення швидких і приблизних обчислень ми можемо скористатися тими ж значеннями a і b , які використовували для закону Фітса .

При використанні будь-яких позитивних і ненульових значень a і b із закону Хіка слід , що надання користувачеві відразу декількох варіантів одночасно зазвичай є більш ефективним , ніж організація тих же варіантів в ієрархічні групи . Вибір з одного меню , що складається з 8 елементів , виробляється швидше, ніж з двох меню , що складаються їх 4 елементів кожне. Якщо всі елементи можуть бути обрані з рівною імовірністю і якщо не враховувати час, необхідний для відкриття другого меню (яке, звичайно , ще більш збільшило б час для інтерфейсу , що складається з двох меню) , то порівняння часу для вибору одного елемента з восьми ($a + b \log_2 8$) з подвоєним часом для вибору одного елемента з чотирьох $2 (a + b \log_2 4)$ покаже , що

$$a + 3b < 2 (a + 2b)$$

оскільки $\log_2 8 = 3$, а $\log_2 4 = 2$, а також оскільки $a < 2a$ і $3b < 4b$. Це узгоджується з даними , отриманими в експериментах зі структурами меню.

Підсумок

ЗНАТИ: особливості кількісного аналізу інтерфейсу.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Як визначається інформаційна продуктивність інтерфейсу?
2. Для чого призначена модель GOMS?
3. Що розраховує закон Хіка?
4. Що розраховує закон Фітса?
5. Охарактеризуйте символну продуктивність інтерфейсу?

Додаткові навчальні матеріали

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Джерела інформації

1. Человеко-машинное взаимодействие: Учебное пособие / Новосибирский государственный технический университет. – Новосибирск, 2006. – 97 с.

Глосарій, персоналій

GOMS; закон Фітса; закон Хіка; інформаційна продуктивність

Лекція №16. Стандарти по проектуванню інтерфейса користувача. Проблеми і тенденції розвитку людино-машинних інтерфейсів

План:

1. Деякі визначення
2. Стандарти по проектуванню інтерфейса користувача
3. Проблеми і тенденції розвитку людино-машинних інтерфейсів

Конспект лекції

16.1 Деякі визначення

Юзабіліті

Це властивість системи або продукту. Якщо продукт є юзабельним, то це означає, що користувачі можуть досягти свої цілі при використанні цього продукту; це означає, що не існує бар'єрів у вирішенні завдань. Фахівці з юзабіліті враховують у своїй роботі ці цілі і завдання. Визначення юзабіліті відповідно до стандарту ISO 9241-11 звучить так: «юзабіліті - це ступінь ефективності, продуктивності і задоволеності, з якими продукт може бути використаний певними користувачами в певному контексті використання для досягнення певних цілей» (the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use). У той же час юзабіліті - це дисципліна, яка займається проектування інтерфейсів для ПО, HCI є науковим фундаментом для юзабіліті.

HCI (human-computer interaction, людино-комп'ютерна взаємодія). Це наукова і прикладна дисципліна, предметом якої є те, як люди використовують комп'ютери і як слід розробляти комп'ютерні системи, для того, щоб забезпечити більш ефективне їх використання кінцевими користувачами. Дисципліна включає елементи психології, ергономіки, інформатики, графічного дизайну, соціології та антропології.

UCD (user-center design, проектування, орієнтоване на потреби користувачів). Підхід до проектування комп'ютерних систем (додатків, веб-сайтів), при якому на перше місце ставляться користувачі, їх потреби і вимоги. При використанні даного підходу досягається високий рівень юзабіліті, що позначається наприклад, на збільшенні продажів (в електронному магазині).

IA (information architecture, інформаційна архітектура). Сукупність методів і прийомів організації інформації для полегшення виконання людьми їх інформаційних потреб з пошуку і перегляду інформації.

Існує кілька загальновизнаних стандартів по юзабіліті і доступності, які застосовуються в проектуванні інтерфейсів. Деякі з них носять рекомендаційний характер, в той час як інші є обов'язковими. При цьому характер вимог може залежати від країни, для якої створюється продукт або від типу системи. Наприклад, компанія, яка розробляє корпоративну систему, пред'являє до свого продукту більш жорсткі вимоги, ніж веб-студія, яка проектує електронний магазин. Компанії, що створюють ПО і веб-сайти для широкої публіки абсолютно зобов'язані дотримуватися стандартів і рекомендацій.

16.2 Стандарти по проектуванню інтерфейса користувача (ISO — International Standardization Organization)

Розташована в Женеві, Міжнародна організація по стандартизації - неурядова мережа національних інститутів по стандартизації, в якій беруть участь представники з

146 країн. Стандарти, які випускає ця організація, формують своєрідний технологічний словник, за допомогою якого виробники продуктів можуть розмовляти один з одним на одній мові. Фактично, ці стандарти є необхідною умовою як для виробництва продуктів, так і для ведення бізнесу. Адже часто для виробництва продукту потрібно безліч сторонніх деталей, які повинні бути сумісні між собою. Знайти ці комплектуючі, замовити їх виробництво або зробити без допомоги загальноприйнятих стандартів попросту неможливо.

Існує кілька **стандартів**, розроблених в ISO і належать до сфери юзабіліті і людино-комп'ютерного взаємодії:

- ISO 9241 - містить вимоги до ергономіки візуальних дисплейних терміналів для офісної роботи. Основний акцент ISO 9241 зроблений на вимоги до офісного обладнання, які повинні виконуватися всіма виробниками, наприклад, вимоги до дисплеїв, клавіатур, до відбиття, до кольору, до компоновання елементів на екрані, до діалогів і повідомлень про помилки. Цей стандарт не застосовний до проектування інтерфейсів для мобільних пристроїв. 11 секція сама велика і загальна. У ній дається визначення юзабіліті - «ефективність, продуктивність і задоволення користувача», а також наводяться різні метрики, які допомагають проектувати зручні інтерфейси.
- ISO 13407 - описаний процес проектування інтерактивних систем, орієнтованих на користувачів. Цей стандарт містить рекомендації щодо організації процесу проектування інтерфейсів і органічному встраюванню цього процесу в загальний процес виробництва ПО. У стандарті описані методи юзабіліті, необхідні для: визначення контексту використання продукту, виявлення вимог користувачів і замовників до системи, прототипування і юзабіліті-тестування продукту.
- ISO 18529 - ергономіка людино-комп'ютерного взаємодії - опис процесу проектування інтерфейсів, орієнтованих на користувачів
- У стандарті детально описана модель зрілості організації з точки зору рівня використання в ній UCD-процесу. Даються рекомендації по переходу на більш високі рівні зрілості.
- ISO 14915 - ергономіка програмного забезпечення мультимедійних користувацьких інтерфейсів. У стандарті даються рекомендації по створенню елементів управління для мультимедійних продуктів, таких, наприклад, як навчальних систем, довідкових кіосків, електронних довідок.
- ISO 16071 - ергономіка взаємодії "людина-система". Настанови щодо доступу до інтерфейсів "людина-машина"
- ISO 16982 - ергономіка взаємодії людина-система. Методи, засновані на зручність застосування, для забезпечення проектування, орієнтованого на людину
- ISO 20282 - юзабіліті повсякденних речей. У першій частині стандарту розповідається про метод визначення властивостей контексту, в якому буде використовуватися розробляється продукт. У другій частині описується методика вимірювання юзабіліті для повсякденних речей.

EU Design для всіх

Суспільство "eEurope initiative" було засновано Європейським союзом в 1999 році для того, щоб «донести вигоди від сучасних технологій до всіх жителів Європи». Це товариство займається інформуванням організацій, що працюють в сфері електронної комерції і контенту, про нові закони, що приймаються в EU і пов'язаних у тому числі з юзабіліті і доступністю.

508 стаття

Це закон уряду США, який вимагає, щоб сайти федеральних установ США були доступні всім, в тому числі і людям з обмеженими можливостями. Він добре виконується і деякі бачать його як модель для закону в Євросоюзі.

Рекомендації уряду Великобританії

Кабінет уряду Великобританії випустив рекомендації для урядових сайтів. Вони в основному базуються на чекліст WAI.

Акт про дискримінацію людей з обмеженими можливостями

Ця державна міра прінімалась в законодавчу базу Великобританії поступово. Вона відноситься до будь-яких організацій, які постачають товари, послуги або обладнання за гроші або безкоштовно. У жовтні 1999 року, сюди включили і інформаційні послуги, пропоновані через Інтернет. Цей закон означає те, що будь-який, хто відчуває, що був дискримінований через те, що виконавець не прийняв «необхідних заходів» для задоволення людей з обмеженими можливостями, може подати на виконавця до суду. Акт не має на увазі конкретно веб-сайти, але є загальна згода в тому, що сайти однозначно підпадають під дію цього закону.

У DDA немає будь-яких рекомендацій. Основний поборник цього закону - RNIB випускає рекомендації, які засновані на чекліст WAI. Їх можна придбати у власність.

WAI (Web Accessibility Initiative, Програма по організації доступу до Мережі для людей з обмеженими можливостями)

Всесвітній Веб Консорціум (W3C - Worldwide Web Consortium) складається з 450 організацій. Мета цього руху - максимально ефективно використовувати потенціал мережі Інтернет. Основна діяльність цієї організації - розробка стандартів. WAI - визнаний стандарт консорціуму, в якому декларується доступність веб-сайтів для всіх людей, незалежно від використовуваної ними платформи або недоліків.

У цьому стандарті містяться рекомендації по доступності в трьох сферах: веб-контенту, засобів для створення контенту і користувацьких агентів (браузери і подібне ПО). До першої сфери відноситься стандарт "Рекомендації загальнодоступності веб-контенту" (WCAG, Web Content Accessibility Guidelines 1.0), який був вперше опублікований в 1999 році. У рекомендаціях наводиться таблиця контрольних перевірок, розбита на три групи за рівнем впливу на загальнодоступність. Залежно від кількості виконаних правил, веб-сайту присвоюється одна з трьох ступенів відповідності: "А", "АА" і "ААА" (ступінь "А" означає, що сайт задовольняє основним вимоги і має низький рівень доступності, а ступінь "ААА" означає 100% доступність сайту).

Дотримання цих рекомендацій носить добровільний характер. Їх виконання підвищує рівень доступності сайту і як наслідок збільшує його аудиторію.

16.2 Проблеми і тенденції розвитку людино-машинних інтерфейсів

Технічні досягнення ХХ століття в області зв'язку й інформації призвели до того, що у світі склалося особливе віртуальне середовище взаємодії людей за допомогою комп'ютера. У зв'язку із цим актуальними стають проблеми ефективної комунікації між людиною та комп'ютером які на даному етапі намагаються вирішити як комп'ютерні інженери, програмісти, так і лінгвісти всього світу. Так, авторами досліджень взаємодії «людина-комп'ютер» є Д.Ятес, В.Орликовський, Д.Гумперт, Т.Нільсен, Л.Спроул, С.Кейслер, С.Хилтз, М.Туроф, Д.Норман та ін.. Питання проблем взаємодії вербальних та невербальних чинників в такій комунікації заторкують в своїх роботах Р.Райс, Д.Лав, Л.Спроул, Д.МакНіл, Д.Росіні, А.Малдер, А.Кендон та ін.. Зараз, в період загальної комп'ютеризації та глобальної інформатизації, на перше місце виходить проблема створення такого інтерфейсу користувача, який не тільки б задовольняв усі вимоги до ефективної взаємодії людини з комп'ютером, а й значно б її полегшував та покращував результативність завдяки якнайбільшій відповідності програмного забезпечення природній мові людини (у її як вербальному, так і невербальному вираженні) та його відповідності всім особливим потребам користувача.

Відомий спеціаліст в галузі комп'ютерної семіотики Д.Норман вважає, що для створення якісного інтерфейсу користувача комп'ютера необхідно володіти: 1) мовами програмування та інформаційних технологій; 2) знанням про людей, особливості їхнього спілкування між собою та з комп'ютерною системою; 3) знанням тієї прикладної галузі,

зادля вирішення завдань якої створюється програмний продукт [1, с.149]. При вивченні інтерфейсу користувача слід брати до уваги природу взаємодії людини з комп'ютерною системою, а також особу самого користувача: його знання, нахили, здібності, потреби, тощо.

З 1988 року в науковій літературі починають обговорюватися основні положення нової науково-практичної спеціалізації, що займається дослідженням проблем, спільних для семіотики, лінгвістики та ергономіки, – ергономічної семіотики (надалі ергосеміотики). Вона спрямована на розробку принципів проектування та створення ефективних та прийнятних мов людино-машинної, зокрема людино-комп'ютерної взаємодії. В основу ергосеміотики покладено ідеї Ч.С.Пірса про те, що розуміння природи іконічних та індексальних знаків та символів може сприяти належному використанню властивих людині мов у створенні штучних знакових систем. Більшість властивих людині кодів з'явилась ще у довербальному періоді еволюції засобів спілкування. До них можна віднести такі знакові системи як мова рухів, дій, жестів, образів, емоційних станів, іконічна мова, музичний та кольоро-коди тощо. У зв'язку з тим, що ці мови та коди не є добре вивченими, до завдань ергосеміотики належать дослідження цих семіотичних утворень у двох напрямках: а) проектування складних семіотичних утворень; б) сполученості якостей, функцій та особливостей штучних мов людино-комп'ютерної взаємодії з характеристиками нових засобів та приладів для введення та виводу інформації.

Останнім часом активно розвиваються засоби спілкування у віртуальній реальності, Інтернеті тощо. На думку Г.Я.Узилевського, цей період відрізнятиметься безпосередньою взаємодією користувачів із комп'ютерами, коли мови та коди, що з'явилися у довербальному періоді, усний варіант природної мови та нові засоби введення та виводу інформації будуть відігравати домінуючу роль; з'являтимуться нові знакові засоби, побудовані на симбіозі цих мов та кодів [4, с.5]. В контексті ергосеміотики конкретний інтерфейс користувача визначається як інтерактивна багаторівнева інформаційна система, утворена на основі вже наявних знакових систем (природної мови, іконічної мови, кольорових та музичних кодів тощо), а також штучних мов, що еволюційно розвиваються, та гнучких способів створення програмних і технічних засобів. Для вдосконалення комунікації «людина-комп'ютер» в її основу має бути закладена не лише модель обробки інформації в процесі спілкування людина-людина, а й модель породження рухів та дій як елементи вербального та невербального мислення, тісно пов'язані між собою прямими та зворотними зв'язками, багатоярусністю та системною організацією. Така модель сприяє введенню в людино-комп'ютерну взаємодію нових метафор, проектуванню сучасних комп'ютерних систем та дослідженню ментального світу користувача.

Характерною рисою розвитку сучасних комп'ютерних технологій є активне впровадження мультимедіа технологій в інформаційні системи як для загальних завдань представлення інформації, так і для забезпечення взаємодії користувача з програмним середовищем. В умовах, коли комп'ютерні технології стають доступними дедалі ширшому колу людей, найбільш перспективним стає напрямок комп'ютерного діалогу з елементами інтелектуального інтерфейсу, який використовує як природну мову, так і невербальні засоби комунікації. Застосування невербальних мовних засобів в спеціалізованих програмних середовищах дозволяє підвищити ефективність їх використання. Вдосконалений таким чином комп'ютерний діалог розширює сигналізаційні можливості інтерфейсу як кількісно, так і якісно. Інтеграція невербальних компонент комунікації у людино-машинний інтерфейс сприяє підвищенню швидкості та повноти засвоєння інформації користувачами програм, що є особливо важливим для таких класів програмного забезпечення, як системи контролю даних, комунікаційні програми, операційні системи, навчальні, розвиваючі та довідкові системи.

З 90-х років минулого століття розпочалися дослідження нового покоління інтерфейсів користувача, які не використовують меню, форми та панелі інструментів; замість них для визначення специфікацій операцій робиться наголос на навчальні приклади, жести та розпізнавання мовлення. Такі інтерфейси ще не набули широкої популярності, проте спеціалісти різних прикладних галузей активно працюють над їх вдосконаленням. Виходячи з цього, в нашому дослідженні ставимо за мету висвітлення прикладних аспектів кінесики – однієї із центральних галузей невербальної семіотики, бо саме вона зараз проходить процес найбільшої інтеграції у людино-машинну комунікацію.

Для вирішення глобальної проблеми людино-машинної взаємодії в інтерфейсах користувача дедалі більше використовуються додаткові види каналів передачі інформації (мовлення, артикуляція губ, жести, напрямок погляду тощо). Такий спосіб взаємодії отримав назву «мультимодальна взаємодія»; вона реалізується шляхом мультимодальних інтерфейсів. Саме такі інтерфейси є властивими міжлюдському спілкуванню. В таких програмах користувач обирає, який канал, для передачі якого типу інформації йому найбільш зручно використовувати у даний момент. Такі інтерфейси дозволяють забезпечити найбільш ефективну та природну для людини взаємодію з різноманітними автоматизованими засобами керування та комунікації. В мультимодальних системах інформація від аудіо, відео, тактильних та інших комунікативних каналів безперервно обробляється, утворюючи реальне чи віртуальне оточення, що дозволяє задовольнити бажання користувача та оперативно адаптуватися до контексту [2, с.301]. Отже, моделювання мультимодальної поведінки людини є основою для проектування інтелектуальних систем природної взаємодії. Мультимодальні мовленнєві системи (аудіовізуальні) є спробою досягнути простоти комунікації «людина-комп'ютер», поєднуючи автоматичне розпізнавання мовлення з іншими невербальними засобами, а також інтегруючи невербальні засоби із синтезом мовлення задля покращення методу введення інформації в мультимодальний пристрій. Таким чином, застосування мультимодальності дозволяє розширити спектр користувачів системи та потенційно забезпечує адаптивність до специфічних умов функціонування.

Можливість за допомогою мультимодальних систем обробляти вхідні модальності паралельно сприяє покращенню ефективності взаємодії людини та комп'ютера, особливо при маніпулюванні графічною інформацією. Оскільки користувачі сприймають інформацію на екранах комп'ютерів візуально (та частково на слух), і використовують рухи для проведення найпростіших операцій у графічному інтерфейсі комп'ютера (такі як клік, подвійний клік, пересування, що супроводжуються виконанням певних команд), тому можна стверджувати, що оптико-кінетичні засоби комунікації превалюють у спілкуванні людини з комп'ютером. Говорячи про рухи у спілкуванні з комп'ютером, варто розглянути природні жести людини в рамках функцій, які вони виконують. На сьогоднішній день існує багато різних класифікацій жестів, запропонованих такими дослідниками невербальної комунікації як C.Cadoz, D.McNeill, D.Efron, P.Ekman, B.Rimé і L.Schiaratura, A.Kendon та ін.; результати їх досліджень висвітлив у своїй роботі A.Mulder [6]. Стосовно застосування жестів у комунікації «людина-комп'ютер», на нашу думку, заслуговує на увагу класифікація, запропонована Д.МакНілом. Він виділяє жести: іконічні (що відображають певний об'єкт чи подію та тісно співвідносяться з семантичним змістом мовлення), метафоричні (як іконічні, але відображають абстрактні поняття), дейктичні (що вказують на когось чи щось конкретне чи то абстрактне) – це жести-вказівки, що застосовуються для привертання уваги, о жести-акценти чи жести-наголоси (що мають лише дві фази (вгору/вниз, всередину/назовні), наголошуючи на значущості слова чи фрази, яких вони стосуються).

Говорячи про взаємодію людини з комп'ютером, в першу чергу згадують саме про дейктичні жести, що не дивно через те, що ці жести використовуються фактично у всіх доменах людино-машинної взаємодії (віртуальна реальність, мобільні телефони, планшети тощо). В роботі R.Bolt "Put that there" (1980) дейктичні жести було використано разом із

командами вводу голосом для вказівки на розташування предметів на великому екрані. Ця робота є одним із перших досліджень з мультимодальних систем, в якому демонструвалася концепція використання мовлення та ручного маніпулювання об'єктом. Починаючи з цієї роботи, почали з'являтися бібліотеки та алгоритми для підтримки нових видів взаємодії, а також реальні програми (додатки). З базової концепції R.Bolt розвинулася загальна ідея мультимодальних систем, а емпірична робота з їх створення сприяла розробці теорії HCMI (human-computer multimodal interaction), яка наразі забезпечує основу для написання посібників з проектування таких систем. Варто також розглянути класифікацію жестів, запропоновану F.QUEK, в якій виділяється п'ять основних типів жестів у взаємодії «людина-комп'ютер»: жестикуляційні, маніпуляційні, сигналізаційні, дейктичні та мовоподібні.

Зупинимось на різноманітних маніпуляційних жестах, які існують у взаємодії людини з комп'ютером. За визначенням F.QUEK, призначенням маніпуляційного жесту є керування певним об'єктом із використанням зв'язку між фактичним рухом рук та маніпуляцією об'єктом [7]. Маніпуляція відбувається як із використанням певних пристроїв для маніпуляції (наприклад, комп'ютерна мишка, стилус), так і з використанням вільних рухів рук для імітації певних дій із фізичним об'єктом; як в інтерфейсах віртуальної реальності, так і в «фізичних» інтерфейсах. До маніпуляційних жестів відносяться жести-вказівки, двовимірні та тривимірні жести. Для вказування використовуються спеціальні указки, палець руки чи світлове перо та активний (сенсорний) екран. Двовимірні жести є рухами на площині, коли маніпулюють безпосередньо курсором, вікном чи іншим об'єктом на екрані. Традиційно для маніпулювання у графічних інтерфейсах користувача використовується мишка, стилус, джойстик, трекбол чи інші засоби. Тут слід зауважити, що такі дії як перетягування та клік не вважаються жестами, доки користувач не виконає певний жест, який інтерпретуватиметься як команда. Тривимірні жести є результатом руху пальців, рук чи голови у тривимірному просторі.

Для створення систем інтерактивної взаємодії «людина-комп'ютер» важливі також класифікації, в яких враховано зв'язок між мовленням та жестами. Так, A.Kendon запропонував розташувати жести в порядку залежності від зв'язків мовлення/жест таким чином:

- жестикуляція (жести наголосу, жести-зв'язки);
- мовоподібні жести (іконічні/образотворчі жести);
- міміка, пантоміма (пантомімічні жести);
- дейктичні жести (вказівні);
- мова жестів (мова глухонімих).

Чим ближче до останньої групи (мова жестів), тим менше необхідність у супроводжувальному мовленні. Жести стають більше подібними до мови. Так, наприклад, мова глухонімих враховує більшість семантичних та синтаксичних особливостей мовлення, тому не потребує самого мовлення для її інтерпретації. Проте іконічні жести (образотворчі) важко зрозуміти без додаткової інформації.

Зупинимось докладніше на жестикуляції та міміці, представлених у класифікації. У взаємодії людини з комп'ютером жестикуляція, як найбільш природна форма жестів, застосовується у мультимодальних інтерфейсах, де мовлення та жести розглядаються як єдиний засіб комунікації. Оскільки всі люди при мовленні так чи інакше застосовують жестикуляцію, актуальним постає її врахування при створенні анімаційних героїв чи роботів, за посередництвом яких відбувається комунікація.

На думку В.Є.Тимофєєва, моделювання на комп'ютері саме мімічного апарату людини є найбільш візуально ефективним засобом удосконалення людино-машинного інтерфейсу. Мова міміки є досить складною та здатна передавати великі обсяги змістовно насиченої та різноманітної інформації. Тому одним із завдань лінгвістів, які займаються вивченням цього питання, є надання розробникам програмного продукту наукової бази

для створення мімічного інтерфейсу, що може стати початком формування комп'ютерної моделі співрозмовника, яка інтегруватиметься у людино-машинний інтерфейс та включатиме як міміку, так і інші невербальні мовні засоби, а також мовлення [3]. Необхідно зауважити, що мімічний компонент у комунікації, напевно, вперше відобразився у створенні та широкому застосуванні «смайликів», що моделюють людські емоції, без яких деякі користувачі вже не можуть повноцінно сприймати певну інформацію. Тобто, моделювання мімічних систем почалося саме із створення графічних символів для позначення людських емоцій, які надалі систематизуються у бази даних, що використовуються для візуалізації міміки. Сучасні ж комп'ютерні програми вже пропонують нам спілкування не просто із зображенням людини на екрані, а з її «оживленим» варіантом, який рухає губами, бровами, головою, має певний вираз обличчя та здатен виражати різні емоції. Одним із найбільш перспективних напрямків досліджень у цій галузі стане трансформування міміки довільного людського обличчя, що вводиться в систему як графічний об'єкт безпосередньо в процесі її роботи. Актуальним є також застосування мімічної системи у навчальних програмах, а саме для оцінювання успішності, тобто такого моделювання функцій вчителя, без якого важко розраховувати на ефективність навчання.

Іноземні дослідники (W.Gao, C.Maggioni, N.Hataoka, K.Bohm та ін.), які вивчають мультимодальну взаємодію, що включає не лише жести та мовлення, але й вирази обличчя (міміку), створили систему, спроможну ідентифікувати кожну модальність окремо. З точки зору дослідників людської поведінки, недоліки такої системи полягають у тому, що при перевірці та зіставленні інформації, отриманої окремо з кожної модальності, виявляється, що такі види інформації дублюють одна одну. Доцільним є створення системи, в якій окремі модальності, виражаючи специфічні аспекти комунікації, не лише б підтверджували правильність інтерпретації інших, а й доповнювали б одна одну.

Розглядаючи кінетичний компонент невербальної комунікації людини з комп'ютером, важливо відзначити той факт, що людина має дві руки. Оскільки 90% людей є праворукими, то й більшість інтерфейсів користувача розробляються з урахуванням того, що майже всі дії виконуватимуться саме правою рукою. Проте дослідники у цій галузі (J.Napier, D.Kimura, W.Buxton та ін.) наголошують, що хоча ліва півкуля головного мозку людини (що відповідає за діяльність правою рукою) є важливою при свідомому виборі рухів, пальці лівої руки краще виконують ізольовані рухи, ніж пальці правої. У.Бакстон та Б.Майєрс, в свою чергу, експериментально довели, що залучення обох рук при редагуванні тексту та одночасній навігації по документу сприяє значному покращенню продуктивності виконання завдання та зменшенню витраченого часу, при чому не лише у досвідчених користувачів, а й у новачків. Таким чином учені доводять, що необхідні навички моторики є природними для людини незалежно від міри її обізнаності з комп'ютерними інтерфейсами і з легкістю застосовуються нею в комунікації «людина-комп'ютер».

Кінетичний компонент комунікації також відіграє важливу роль у діалозі з комп'ютером користувачів із вадами зору, для яких вже було винайдено брайлевський дисплей та брайлевську клавіатуру. Проте в цьому випадку ефективнішою стає комбінація аудіальних та тактильних чинників. Для користувачів із вадами слуху нещодавно було сконструйовано спеціальну сенсорну рукавичку, яка здатна перекладати жестову мову у текст на дисплеї.

В ході досліджень було виявлено, що приблизно 95% користувачів надають перевагу мультимодальній взаємодії в роботі, наприклад, із інтерактивними картами, вільно використовуючи як мовленнєве введення, так і вказівки пальцем чи світлове перо. Застосування мультимодальних інтерфейсів сприяє збільшенню швидкості виконання візуально-просторових завдань на 10%.

Найбільші функціональні можливості мають системи, які використовують мультимодальне подання інформації, тобто застосовують, паралельно або послідовно,

кілька різних комунікаційних каналів (слуховий, зоровий, тактильний та ін.). Завдяки цьому мультимодальні системи дають змогу надзвичайно збільшити обсяг передавання інформації, що є головною їх перевагою. Мультимодальність спілкування підвищує ефективність навчання завдяки притаманному людині механізмові паралельного опрацювання інформації, що надходить різними каналами. Потрібно проектувати систему таким чином, щоб всі сенсорні модальності доповнювали, а не дублювали і заважали одна одній, примушуючи переключати увагу і відволікати внутрішні ресурси людини від переробки інформації. Доцільно вважати мультимодальність необхідною умовою психологічної природності та комфортності спілкування.

Розробка системних принципів створення комфортного та ефективного діалогу людини з комп'ютером передбачає вивчення основних компонентів системи «людина-комп'ютер», а саме психологічних, семіотичних, ергономічних та інших аспектів процесу комунікації. В статті ми проаналізували ефективність залучення та використання у сучасних комп'ютерних програмах жестів та міміки задля більш ефективної взаємодії людини з комп'ютером. Незважаючи на те, що окремі дослідники виводять на перший план різні невербальні чинники, аналіз їх застосування у людино-машинному спілкуванні показав, що важливість того чи іншого компонента для користувача є, по-перше, ситуативно-зумовленою, а по-друге, залежить як від особливих потреб користувача (наприклад, певних фізичних вад), так і від кінцевих цілей комунікації.

Зараз рішенням проблем ефективної взаємодії людини з комп'ютером у природній формі займаються спеціалісти різних наукових галузей. Інженери, математики, програмісти розробляють більш ергономічні програмні та апаратні засоби керування комп'ютерами. В свою чергу лінгвісти, фізіологи, психологи вивчають поведінку людини, а також механізми сприйняття, обробки інформації задля того, щоб зрозуміти специфіку процесів мислення. Отже, на нашу думку, на даний час виглядає найбільш перспективним міждисциплінарний підхід до вирішення проблем побудови ефективних та природних інтерфейсів.

Підсумок

ЗНАТИ: особливості кількісного аналізу інтерфейсу.

ВМІТИ: використовувати на практиці при проектуванні інтерфейсу людина-комп'ютер.

Контрольні питання і вправи

1. Для чого потрібні стандарти на інтерфейс користувача?
2. У якому стандарті наведено визначення юзабіліті?
3. Які перспективи голосових інтерфейсів?
4. Охарактеризуйте природно-інтуїтивні інтерфейси.

Додаткові навчальні матеріали

1. Інтерфейс "Користувач-комп'ютер": Навчальний посібник / В.П. Майданюк, А.М. Петух. - Вінниця: ВДТУ, 1999. - 66 с.

Джерела інформації

1. Агеев В. Н. Семиотика. — М.: Весь Мир, 2002. — 256 с.
2. Ронжин А. Л., Карпов А. А. Многомодальные интерфейсы: основные принципы и когнитивные аспекты / А. Л. Ронжин, А. А. Карпов // Труды СПИИРАН. Вып. 3, т. 1.— СПб.: Наука, 2006. — С.300-319.

Глосарій, персоналій

ISO; BCI-інтерфейс; стандарт; голосовий інтерфейс; природно-інтуїтивний інтерфейс;